Proceedings of the BDA 2020 conference

Bernd Amann, François Goasdoué

BDA 2020

—

Gestion de Données
Principes Technologies et Applications

Bernd Amann[1] and François Goasdoué[2]

[1]LIP6 - Sorbonne Université. Bernd.Amann@lip6.fr
[2]IRISA - Université de Rennes 1. Francois.Goasdoue@irisa.fr

Actes de la conférence BDA 2020

Conférence soutenue par

Sorbonne Université, Université Paris-Saclay, INSA Valor et GDR MaDICS.

Site web de la conférence : `https://bda.lip6.fr`

Table des matières

1 Message de la présidente des journées et des organisateurs

La conférence « BDA : Gestion de Données – Principes, Technologies et Applications » est le rendez-vous annuel incontournable de la communauté de la gestion de données en France. Sa 36ème édition a eu lieu du 27 au 29 octobre 2020. Elle devait initialement se dérouler à Paris mais a été finalement organisée en ligne pour cause de crise sanitaire liée au COVID-19.

On estime que le volume de données produites dans le monde double tous les trois ans ! Plus que jamais, on a donc besoin de savoir stocker, indexer, interroger, traiter, exploiter, analyser ces énormes quantités de données, qui se caractérisent par leur diversité, leur complexité, leur volume, leur forte évolutivité. Poursuivant la tradition des rencontres annuelles de la communauté de gestion de données francophone, la conférence BDA a invité les acteurs académiques et industriels à soumettre leurs travaux récents sur les défis et les avancées scientifiques dans leurs domaines, et avec plus de 200 participants, l'édition BDA 2020 atteste une nouvelle fois du dynamisme de sa communauté et de l'importance de la recherche en gestion de données.

Le programme scientifique final comporte 32 articles de recherche dont 29 longs et 3 courts, 7 démonstrations et 2 articles de doctorants. Il a été complété par trois conférences invités sur des sujets d'actualité. Un objectif important de BDA est de donner la possibilité aux chercheurs, et surtout aux doctorants, de présenter leurs travaux à la communauté, ce qui inclut également des travaux récents déjà publiés au moment de la soumission. Ces actes de conférences proposent ainsi des résumés de toutes les contributions publiées et non-publiées et ils sont complétés par une édition spéciale du journal Transactions on Large-Scale Data and Knowledge-Centered Systems (TLDKS) avec les versions étendues d'une sélection d'articles acceptés. Depuis quelques années, BDA attribue des prix pour des contributions exceptionnelles et cette année les comités ont à nouveau récompensé quatre articles de recherche, une démonstration, ainsi que deux thèses indiqués à la fin de ces actes.

Nous tenons encore une fois à remercier tous les auteurs pour la qualité des contributions et des présentations associées, les membres de l'équipe d'organisation, les membres des comités de programme et du prix de thèse. Nous remercions en particulier Meghyn Bienvenu, Melanie Herschel et Jorge Quiané, pour avoir accepté notre invitation et pour l'excellence de leur présentations. Nous sommes également reconnaissants envers nos soutiens qui ont permis d'organiser cette manifestation : le Laboratoire LIP6 de Sorbonne Université, l'Université Paris-Saclay, INSA Valor et le GDR CNRS MaDICS.

Nos remerciements vont enfin aux nombreux participants qui ont fait vivre cette édition BDA 2020 si particulière.

Anne Doucet, Présidente des journées BDA 2020
Bernd Amann, Président du Comité d'Organisation
François Goasdoué, Président du Comité de Programme

2 Comités BDA 2020

2.1 Présidente des journées

— Anne Doucet, LIP6 – Sorbonne Université

2.2 Comité d'organisation

— Bernd Amann, LIP6 – Sorbonne Université (Président)
— Nicole Bidoit, LRI – Université Paris Saclay
— Mohamed-Amine Baazizi, LIP6 – Sorbonne Université
— Camelia Constantin, LIP6 – Sorbonne Université
— Stéphane Gançarski, LIP6 – Sorbonne Université
— Hubert Naacke, LIP6 – Sorbonne Université

2.3 Comité de programme

— François Goasdoué, IRISA, Université Rennes 1 (Président)
— Sihem Amer-Yahia, LIG, CNRS, Université Grenoble Alpes
— Salima Benbernou, LIPADE, Université Paris Descartes
— Laure Berti-Equille, IRD
— Luc Bouganim, INRIA Saclay
— Bogdan Cautis, LRI, Université Paris Saclay
— Sylvie Cazalens, LIRIS, INSA Lyon
— Serenella Cerrito, Université Paris-Saclay, Univ. Evry, Laboratoire IBISC
— Dario Colazzo, LAMSADE, Université Paris-Dauphine
— Camelia Constantin, LIP6, Sorbonne Université
— Amelie Gheerbrant, IRIF, Université Paris Diderot
— Daniela Grigori, LAMSADE, Université Paris-Dauphine
— David Gross-Amblard, IRISA, Université Rennes 1
— Mohand-Said Hacid, LIRIS, Université Claude Bernard Lyon 1
— Allel Hadjali, LIAS, ISAE – ENSMA
— Mirian Halfeld-Ferrari-Alves, LIFO, Université d' Orléans
— Abdelkader Hameurlain, IRIT, Université Paul Sabatier
— Hélène Jaudoin, IRISA, Université Rennes 1
— Frédérique Laforest, LIRIS, INSA Lyon
— Sofian Maabout, LABRI, Université de Bordeaux
— Ioana Manolescu, INRIA Saclay

— Cedric du Mouza, CEDRIC, CNAM
— Benjamin Nguyen, LIFO, INSA Centre Val de Loire
— Nathalie Pernelle, LIPN, Université Paris 13
— Jean-Marc Petit, LIRIS, INSA Lyon
— Olivier Pivert, IRISA, Université Rennes 1
— Philippe Rigaux, CEDRIC, CNAM
— Claudia Roncancio, LIG, Grenoble INP
— Marie-Christine Rousset, LIG, Université de Grenoble Alpes
— Pierre Senellart, ENS DI, ENS
— Virginie Thion, IRISA, Université Rennes 1
— Farouk Toumani, LIMOS, Université Clermont Auvergne
— Federico Ulliana, LIRMM, Université de Montpellier
— Dan Vodislav, ETIS, Université de Cergy-Pontoise
— Karine Zeitouni, DAVID, Université de Versailles Saint-Quentin

2.4 Comité de démonstration

— Nicolas Travers, Ecole Supérieur d'Ingénieurs Léonard de Vinci (Président)
— Tristan Allard, IRISA, Université Rennes 1
— Mohamed-Amine Bazizi, LIP6, Sorbonne Université
— Emmanuel Bruno, LIS, Université de Toulon
— Raja Chiky, ISEP
— Maude Manouvrier, LAMSADE, Université Paris-Dauphine
— Pascal Poncelet, LIRMM, Université de Montpellier
— Fatiha Saïs, LRI, Université Paris Saclay
— Patricia Serrano Alvarado, LS2N, Université de Nantes
— Yehia Taher, DAVID, Université de Versailles Saint-Quentin
— Genoveva Vargas Solar, LIG, CNRS, Université Grenoble Alpes

2.5 Comité du prix de thèse

— Angela Bonifati, LIRIS, Université Claude Bernard Lyon 1 & Inria (Présidente)
— Reza Akbarinia, Inria SAM et Univ. Montpellier 2
— Dario Colazzo, LAMSADE, U. Paris Dauphine
— Pierre Genevès, LIG – Inria Grenoble Rhône-Alpes
— Leonid Libkin, ENS PSL University & Inria
— Themis Palpanas, LIPADE, University of Paris & French University Institute IUF
— Hala Skaf-Molli, LS2N, Université de Nantes
— Farouk Toumani, LIMOS, Université Clermont Auvergne

3 Conférences invitées

3.1 A Guided Tour of Ontology-Mediated Query Answering
Meghyn Bienvenu

Over the past fifteen years, ontology-mediated query answering has grown into a very active research topic within the AI and database theory communities. While enriching data with an ontology offers many advantages (e.g. simplifying query formulation, integrating data from different sources, providing more complete answers to queries), it also renders the query answering task more computationally involved, spurring the development of new algorithmic techniques. The aim of this talk is to provide a gentle introduction to ontologies and ontology-mediated query answering, while also highlighting some recent results and research directions.

Meghyn Bienvenu is a CNRS researcher and member of the LaBRI laboratory at the University of Bordeaux. Born in Canada, she obtained her undergraduate degree from the University of Toronto before moving to France to continue her studies at the University of Toulouse. Her PhD thesis, defended in 2009, was awarded the AFIA Prize for best French dissertation in artificial intelligence. Her research interests span a range of topics in knowledge representation and reasoning and database theory, with a main focus on description logic ontologies and their use in querying data. She currently leads an ANR AI Chair on the topic of intelligent handling of imperfect data. Bienvenu is an associate editor of ACM Transactions on Computational Logic and will serve as PC co-chair for KR 2021, the leading conference on knowledge representation and reasoning. Her research has been recognized by an invited Early Career Spotlight talk at IJCAI'16, the world's premier AI conference, and the 2016 CNRS Bronze Medal in the area of computer science.

3.2 Tackling data quality issues : on getting in shape, playing them offense or defense, and analyzing results
Melanie Herschel

Quality issues are omnipresent in data, the foundation of many business operations and any data analysis. Efforts to address these issues target different steps of the data analysis pipeline. To get data in best shape for further use, methods to avoid wrong data entry or their interpretation are crucial. Processing these data can introduce further quality issues, best to be recognized and rectified. Nevertheless, even with top-notch solutions to mitigate data quality issues, the final result should always be critically examined, which justifies techniques to better understand data-driven results.

In this talk, I will introduce, through a series of anecdotes, the importance and difficulty of data quality in various disciplines and highlight some of our recent contributions to address specific problems at different steps of the data analysis pipeline. The discussion will highlight that poor data quality is and will continue to be a persistent opponent in an increasingly data-driven world, but pushing the limits to improve it is worthwhile, for technical reasons and beyond.

Melanie Herschel is a full professor of Data Engineering at the University of Stuttgart. She was previously an associate professor at Université Paris Sud. In her early career, she was a member of the research staff of the Database Systems group at the University of Tübingen, at IBM Research – Almaden, and at the Hasso-Plattner-Institute Potsdam. She has also held a secondary appointment as Visiting Research Professor at the National University of Singapore. She obtained her PhD from Humboldt University Berlin in 2008. Her research interests in data management include data quality, data integration, meta-data management, and data exploration and analysis. She has participated in the organization of several conferences and workshops, notably as PC chair of EDBT 2019. She is an associate editor for the VLDB Journal and ACM/IMS Transactions on Data Science, and has regularly served as a reviewer for journals and conferences.

3.3 Towards a World-Wide Data Processing Layer
Jorge Quiané

Data science is driven by a large number of data-related assets, such as datasets, algorithms, ML models, and processing systems. Although we all benefit from the latest results in data science, building a proper data science ecosystem requires a significant investment from organizations and individuals. As a result, only a few players can afford such investments, which leads to a small data science "world" dominating the latest technologies. This naturally causes lock-in effects and hinders features that require a flexible exchange of assets among users.

This talk presents Agora, our current effort to "democratise" data science. We are building a unified ecosystem that brings together data, algorithms, models, and computational resources and provides them to a broad audience. In particular, this talk presents the execution layer of Agora. I will talk about a series of works that allow us to : (i) leverage existing execution engines to run tasks efficiently ; (ii) run tasks at different geo-distributed sites without violating the constraints/policies that every site might impose over the data ; and (iii) securely execute tasks at large scale without any data and code (application logics) leakage.

I shall conclude this talk with a roadmap of open problems to achieve a world-wide data processing layer, thereby making a big step forward to fulfil our Agora vision.

Jorge Quiané is Principal Researcher at the DIMA group (TU Berlin) and Scientific Coordinators of the Berlin Institute for the Foundations of Learning and Data (BIFOLD). He is also Scientific Advisor at the IAM group (DFKI). Earlier in his career, he was Senior Scientist at the Qatar Computing Research Institute (QCRI) and Research Associate at Saarland University. Jorge's research interests are in the broad area of scalable data management, including cross-platform data management and big data analytics. He has published numerous research papers on query and data processing as well as on novel system architectures. He also holds 5 patents in core database areas, such as join processing and data storage. He did his PhD in Computer Science at INRIA and University of Nantes, France. He received an M.Sc. in Computer Science with a speciality in Networks and Distributed Systems from Joseph Fourier University, Grenoble, France. He also obtained, with highest honours, an M.Sc. in Computer Science from the National Polytechnic Institute, Mexico.

4 Résumés des articles long

SEPAR: Towards Regulating Future of Work Multi-Platform Crowdworking Environments with Privacy Guarantees

Mohammad Javad Amiri[1] Joris Duguépéroux[2] Tristan Allard[2] Divyakant Agrawal[1] Amr El Abbadi[1]
[1]University of California Santa Barbara, [2]Univ Rennes, CNRS, IRISA
[1]Santa Barbara, California,[2]Rennes, France
{amiri,agrawal,amr}@cs.ucsb.edu,{joris.dugueperoux,tristan.allard}@irisa.fr

Abstract

Crowdworking platforms provide the opportunity for diverse workers to execute tasks for different requesters. The popularity of the "gig" economy has given rise to independent platforms that provide competing and complementary services. Workers as well as requesters with specific tasks may need to work for or avail from the services of multiple platforms resulting in the rise of *multi-platform crowdworking systems*. Recently, there has been increasing interest by governmental, legal and social institutions to enforce regulations, such as minimal and maximal work hours, on crowdworking platforms. Platforms within multi-platform crowdworking systems, therefore, need to collaborate to enforce cross-platform regulations. While collaborating to enforce global regulations requires the *transparent* sharing of information about tasks and their participants, the *privacy* of all participants needs to be preserved. In this paper, we propose an overall vision exploring the regulation, privacy, and architecture dimensions for the future of work multi-platform crowdworking environments. We then present *SEPAR*, a multi-platform crowdworking system that enforces a large sub-space of practical global regulations on a set of distributed independent platforms in a privacy-preserving manner. SEPAR, enforces *privacy* using *lightweight and anonymous tokens*, while *transparency* is achieved using fault-tolerant *blockchains* shared across multiple platforms. The privacy guarantees of SEPAR against covert adversaries are formalized and thoroughly demonstrated, while the experiments reveal the efficiency of SEPAR in terms of performance and scalability.

Reducing the Cost of Aggregation in Crowdsourcing

Rituraj Singh
Univ. Rennes 1
rituraj.singh@irisa.fr

Loïc Hélouët
Inria Rennes
loic.helouet@inria.fr

Zoltan Miklos
Univ. Rennes 1
zoltan.miklos@irisa.fr

ABSTRACT

Crowdsourcing is a way to solve problems that need human contribution. Crowdsourcing platforms distribute replicated tasks to workers, pay them for their contribution, and aggregate answers to produce a reliable conclusion. A fundamental problem is to infer a correct answer from the set of returned results. Another challenge is to obtain a reliable answer at a reasonable cost: unlimited budget allows hiring experts or large pools of workers for each task but a limited budget forces to use resources at best.

This paper considers crowdsourcing of simple boolean tasks. We first define a probabilistic inference technique, that considers difficulty of tasks and expertise of workers when aggregating answers. We then propose CrowdInc, a greedy algorithm that reduces the cost needed to reach a consensual answer. CrowdInc distributes resources dynamically to tasks according to their difficulty. We show on several benchmarks that CrowdInc achieves good accuracy, reduces costs, and we compare its performance to existing solutions.

CCS CONCEPTS

• **Information systems → Crowdsourcing**;

KEYWORDS

Crowdsourcing; Aggregation; Cost; Quality

1 INTRODUCTION

Crowdsourcing is a way to solve tasks that need human contribution. These tasks include image annotation or classification, polling, etc. Employers publish tasks on an Internet platform, and these tasks are realized by workers in exchange for a small incentive [1]. Workers at crowdsourcing platforms are very heterogeneous: they have different origins, domains of expertise, and expertise levels. One can even consider malicious workers, that return wrong answers on purpose. To deal with this heterogeneity, tasks are usually replicated: each task is assigned to a *set of workers*. Each worker executes his assigned task independently and returns his own belief about the answer. As workers can disagree, the role of a platform is then to build a consensual final answer out of the values returned.

A natural way to derive a final answer is **Majority Voting** (MV), i.e. choose as a conclusion the most represented answer. A limitation of MV is that all answers have equal weight, regardless of the expertise of workers. One can easily replace MV by a weighted vote. However, this raises the question of measuring workers expertise, especially when workers competencies are not known a prior. A way to obtain an initial measure of workers expertise is to use **Golden Questions** [4], but this requires additional budget or is not engaging to workers. Several papers have considered aggregation problem and we refer interested readers to [6].

Tasks at crowdsourcing platforms come with a budget. Standard *static* approaches on crowdsourcing platforms fix prior number of k workers per task. The first case is when a client has n tasks to complete with a total budget of B_0 units. Each task can be realized by $k = B_0/n$ workers. The second case is when an initial budget is not known, and the platform fixes an arbitrary number of workers usually between 3 and 10 to each task [2]. An obvious drawback of static allocation of workers is that all tasks benefit from the same work power, regardless of their difficulty. Even a simple question where the variance of answers is high calls for a sampling of larger size. So, one could expect each task t to be realized by k_t workers, where k_t is a number that guarantees that the likelihood to change the final answer with one additional worker is low. However, without prior knowledge on the task's difficulty and on variance in answers, this number k_t cannot be fixed a priori.

In this paper, we first propose an aggregation model based on Expectation Maximization (EM) [3] technique considering factors such as task difficulty and worker expertise to derive an aggregated final answer. Next, we propose CrowdInc, a dynamic worker allocation algorithm that handles at the same time aggregation of answers, and optimal allocation of a budget to reach a consensus among workers. The algorithm works in rounds, forging aggregated answer for tasks at each round. Evaluation for a task stops with a final aggregated answer when this answer has achieved a sufficient confidence.

2 AGGREGATION MODEL

We consider boolean filtering tasks, i.e. tasks with answers in $\{0, 1\}$, but the setting can be easily extended to tasks with any finite set of answers. For each task, an actual ground truth exists, but it is not known by the system. We assume a set of k independent workers, which role is to realize a task, i.e. return an *observed label* in $\{0, 1\}$ according to their belief. We consider a set of tasks $T = \{t_1, \dots t_n\}$ for which a label must be evaluated. For a task $t_j \in T$ the observed label given by worker $1 \leq i \leq k$ is denoted by l_{ij}. Let y_j denote the *final label* of a task t_j obtained by aggregating the answers of all workers. $L_j = \bigcup_{i \in 1..k} l_{ij}$ denotes the set of all labels returned by workers for task t_j, L denotes the set of all observed labels, $L = \bigcup_{j \in 1..n} L_j$. The goal is to estimate the ground truth by synthesizing a set of *final labels* $Y = \{y_j, 1 \leq j \leq n\}$ from the set of *observed labels* $L = \{L_j\}$ for all tasks.

First, we model the *difficulty* of a task j by a real valued parameter $d_j \in [0, 1]$. Value 0 means that the task j is very easy, and $d_j = 1$ means that it is extremely difficult. We model the expertise of a worker i as a pair $\xi_i = \{\alpha_i, \beta_i\}$, where α_i is the **recall** and β_i the **specificity** of worker i. The *recall* α_i is the probability that worker i answers 1 when the ground truth is 1, i.e $\alpha_i = Pr(l_{ij} = 1 | y_j = 1)$.

The *specificity* β_i is the probability that worker i answers 0 when the ground truth is 0, i.e. $\beta_i = Pr(l_{ij} = 0|y_j = 0)$. Next we define a generative function to determine the probability of correct answers by a worker based on defined variables α_i, β_i and d_j. The function is given as $Pr(l_{ij} = y_j|d_j, \alpha_i, y_j = 1) = (1 + (1 - d_j)^{(1-\alpha_i)})/2$ and $Pr(l_{ij} = y_j|d_j, \beta_i, y_j = 0) = (1 + (1 - d_j)^{(1-\beta_i)})/2$.

We use Expectation Maximization (EM) technique to estimate jointly latent variables α_i, β_i, d_j and derive the *final answer* y_j for each task j. EM iterates in two alternating steps. In the E-step, we compute the posterior probability of $y_j \in \{0, 1\}$ for a task j given the difficulty d_j, the workers expertise α_i, β_i and the answers $L_j = \{l_{ij} \mid i \in 1..k\}$. In the M-Step, we maximize the joint log likelihood of the given answers L_j, final answer y_j (computed during the last E-step) to estimate the new value of parameters α_i, β_i and d_j. The algorithm converges, and stops when the difference between two successive joint log-likelihood values is below a threshold. In the end, the algorithm returns the final answer y_j for each task, expertise of the workers α_i, β_i and difficulty of the task y_j.

3 COST MODEL

A drawback of existing crowdsourcing approaches is that task distribution is static, i.e. tasks are distributed to a fixed number of workers, without considering their difficulty, nor checking if a consensus can be reached with fewer workers. We propose CrowdInc algorithm with a dynamic worker allocation strategy to optimize cost and accuracy. CrowdInc algorithm works in rounds, and we denote by q the current round number.

CrowdInc starts with the *Estimation* phase and allocates k workers for an initial evaluation round ($q = 0$). After collection of answers, and then at each round $q > 0$, we first apply EM based aggregation to estimate the difficulty d_j^q of each of task t_j, the confidence $\hat{c}_j^q$ in final aggregated answer y_j^q, and the expertise α_i^q, β_i^q of the workers. The confidence $\hat{c}_j^q$ denotes the collective belief of the workers in the final answer y_j^q. Then a stopping threshold Th^q is used to decide whether there is a need for more answers for each task. If $\hat{c}_j^q$ is greater than Th^q, the task t_j is removed from the existing set of executable tasks. This stopping criterion hence takes a decision based on the confidence in the final answers for a task and on the remaining budget. Once solved tasks have been removed, we compute the number $\mathbf{a}_j^q$ of workers to assign each remaining task t_j following a difficulty aware policy such that more difficult tasks (i.e. with the more disagreement) are allocated more workers than easier tasks. Note that, each task gets a different number of workers based on task difficulty. The algorithm stops when either the whole budget is exhausted or there is no additional task left.

4 EXPERIMENTS

We evaluate the algorithm on three public available dataset and all tags appearing in the benchmarks were collected via Amazon Mechanical Turk [5]. We found that the proposed aggregation technique outperforms the existing techniques. We refer the interested readers to the full paper [5] for complete results. In the next experiment, we verify that the CrowdInc achieves at least the same accuracy but with a smaller budget. The results are given in Figure 1. Static(MV) denotes the results obtained with traditional crowdsourcing platforms with majority voting as the aggregation technique and Static(EM) shows the results obtained with advanced aggregation technique with EM based aggregation technique.

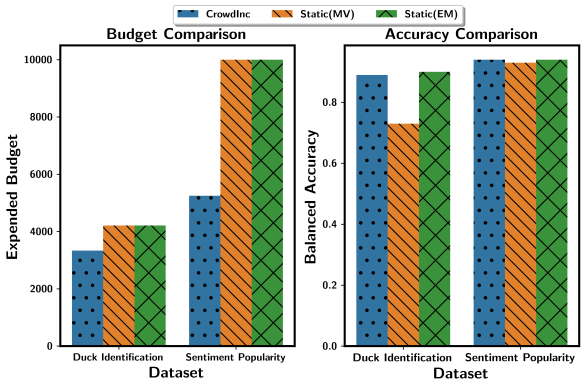


Figure 1: Comparison of cost vs. Accuracy.

The following observation can be made from Figure 1. CrowdInc achieves better accuracy than a static(MV) approach and almost the same accuracy as a Static(EM). Furthermore, CrowdInc is able to achieve at least the same accuracy at a reduced cost.

5 CONCLUSION

In this paper, we introduced an aggregation technique for crowdsourcing platforms. Aggregation is based on expectation maximization and jointly estimates the answers, the difficulty of tasks, and the expertise of workers. We also proposed CrowdInc an incremental labeling technique that optimizes the cost of answers collection. The algorithm implements a worker allocation policy that takes decisions from a dynamic threshold computed at each round, which helps to achieve a trade-off between cost and accuracy.

The work can be extended in several directions. For simplicity, we considered the boolean tagging tasks, but the algorithm can be extended to tasks with a finite number of answers. Another possible improvement is to try to hire experts when the synthesized difficulty of a task is high, to avoid hiring numerous workers or increase the number of rounds. Last, we think that the complexity of CrowdInc can be improved. An interesting idea is to consider how a part of computations can be reused from around to the next one to speed up convergence.

REFERENCES

[1] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh. 2018. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques, and Assurance Actions. *Comput. Surveys* 51, 1 (2018), 7.

[2] H. Garcia-Molina, M. Joglekar, A. Marcus, A. Parameswaran, and V. Verroios. 2016. Challenges in data crowdsourcing. *Trans. on Knowledge and Data Engineering* 28, 4 (2016), 901–911.

[3] M.R. Gupta and Y. Chen. 2011. Theory and use of the EM algorithm. *Foundations and Trends in Signal Processing* 4, 3 (2011), 223–296.

[4] J. Le, A. Edmonds, V. Hester, and L. Biewald. 2010. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*, Vol. 2126. 22–32.

[5] Rituraj Singh, Loïc Hélouët, and Zoltan Miklos. 2020. Reducing the Cost of Aggregation in Crowdsourcing. In *International Conference on Web Services*. Springer, 77–95.

[6] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proc. of the VLDB Endowment* 10, 5 (2017), 541–552.

Temporal Aggregation of Spanning Event Stream: A General Framework

Aurélie Suzanne
aurelie.suzanne@ls2n.fr
Université de Nantes
Nantes, France
Expandium
Saint-Herblain, France

Guillaume Raschia
guillaume.raschia@ls2n.fr
Université de Nantes
Nantes, France

José Martinez
jose.martinez@ls2n.fr
Université de Nantes
Nantes, France

CCS CONCEPTS

• **Information systems → Stream management**.

KEYWORDS

data stream, spanning events, temporal aggregates, temporal database, window query

1 INTRODUCTION

The Big Data era requires new processing architectures among which the stream systems that have become well-known. Those systems are able to summarize infinite data streams with aggregates on the most recent data, allowing keeping a limited but meaningful piece of the initial stream. However, up to now, only point events have been considered and spanning events, which come with a duration, have been let aside, restricted to the persistent databases world only. In this paper, we propose a unified framework to deal with such stream mechanisms on spanning events.

2 EXAMPLE

Let us consider a network monitoring system where we want to evaluate the load of an antenna, with spanning transactions, e.g., phone calls, happening continuously. In a classical streaming system, the load would be based either on the start or end time of the event. With a spanning event stream the full event duration would be interpreted.

Figure 1 models a series of calls: events a_i as point events show only their ending time, while b_i's as spanning events show the full-call duration. We want to analyze the load of the antenna every five minutes showed by windows W_i's. With spanning events, window w_2 contains 4 events: $\{b_3, b_4, b_5, b_7\}$, while point events would find only 2 events for the same window $\{a_3, a_4\}$. This results in more accurate results for spanning events.

Of course, it would be possible to handle both start and end times for each event and then, mimic the spanning event behavior with current streaming systems. However, this would come with some problems to solve: how to deal, for instance, with long-standing events? Or lost messages (never-ending or un-started events)? Or even reversed bound messages (end, then start timestamps)? Moreover, natively modeling event duration allows detecting events which have no bounds in the window, like event b_7 crossing window w_2 and w_3 on Figure 1. Spanning event stream hence allows getting not only information about (dis)connections to/from the antenna, but also to the full connection information.

Spanning events stream hence allows to modelize events in a similar way than the way they were in the real world, providing more accurate results than point events stream. Furthermore reproducing spanning event behavior with current streaming systems would be tricky and time consuming, thus the need for a specific system. In this paper, we propose a unified framework to deal with such stream mechanisms on spanning events.

3 SPANNING EVENT STREAM

A spanning event stream contains a possibly infinite number of spanning events. Those events are composed of a transaction time, a valid time and some data specific to the event.

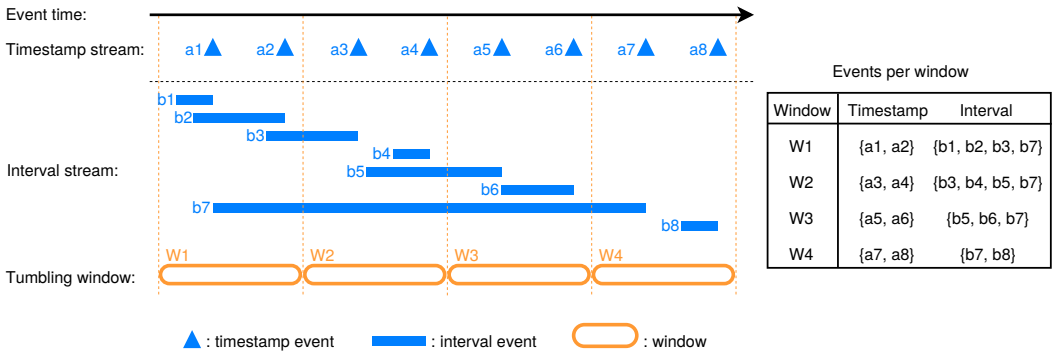
The transaction time corresponds to the moment when the event was received in the system, while the valid time is an interval corresponding to when the event was happening in the real world. Basically, a spanning event is a point event which valid time has been changed to an interval instead of a timestamp.

4 WINDOWING

A common solution to overcome the infinite stream problem with blocking operators, like aggregations, is to use windowing. Indeed, windows extract from the infinite stream finite sub-stream to feed the aggregation operators, which can then calculate results.

Those windows can be represented with intervals, and they are created with measures which set their size and frequency. Those measures can be independent of the stream, like a system clock stored on the streaming system server, or depend on the various parts of the events.

Windows are created with a function, which takes as input one or several measures and output intervals. Then, a predicate is used to fill them. It compares the event valid or transaction time with the window interval to tell if an event is in the window or not. For this, two predicates can be used, the first one comparing an event timestamp to a window interval, the second one using Allen's algebra to compare two intervals.



Events per window

Window	Timestamp	Interval
W1	{a1, a2}	{b1, b2, b3, b7}
W2	{a3, a4}	{b3, b4, b5, b7}
W3	{a5, a6}	{b5, b6, b7}
W4	{a7, a8}	{b7, b8}

Figure 1: End time vs. full-time events aggregation in a window-based stream system.

5 ASSIGNING SPANNING EVENTS TO WINDOWS

With the definition of spanning event stream and windows, the impact of spanning events can now be clearly studied. A first remark we should have is that normally a window is released as soon as its upper bound is reached, but spanning events duration makes to us no guarantee that we will receive an event before it ends (and in fact we might often receive it only when it ends). Hence we need to wait an additional time after the window has closed to actually release the window. We call this time a Time-To-Postpone, and it can as much be learned by the system as given by the user.

Once we know when to release the window, we still need to study precisely how spanning events impacts windows. For this we study here only the most common windows. Among them sliding windows is a good candidate. Those windows advance with the time, number of events received or data. Basically only the sliding windows using time are impacted as they use the valid time of the events to insert them into windows. Hence for them we need to use Allen predicates which compare two intervals together.

Then we can have some look at session windows. Those windows open with the arrival of an event and close when no event has been received for a certain duration. With spanning events we propose to adapt those windows to consider not only the end bound of the event, but the full event duration as being in the session. However we need to keep care of really long events, which could lead to two impossible choices: reopening windows which have already been released, or having overlapping session bounds. To avoid this, we limit the event size to the Time-To-Postpone which makes us sure to avoid clashes between the sessions which need to be created and those which have been released.

6 EXPERIMENTS

Eventually, we validate the soundness of our new framework with a set of experiments, based on a straightforward implementation. In those experiments we show that not only spanning events are more accurate than point events, but the Time-To-Postpone is also a good measure to deal with event duration. This Time-To-Postpone has on top only a limited impact on throughput. When comparing throughput of points events and spanning events on a real-like data set, we acknowledge a loss with spanning events which is, however, low and can be compensated with further optimization techniques.

7 CONCLUSION

In conclusion, spanning events can be used in streaming system, with a gain in accuracy for aggregation results. Indeed, they allow us to modelize the events the same way they were in the real world without making any truncation. However using spanning events implies that we modify the way we use windows. For this, the bound function needs to be adapted, in particular for session windows to acknowledge for the full event duration. Then, the insertion predicate needs to be adapted to compare two intervals together instead of asserting that a timestamp is in an interval. Finally, we need to postpone the window release to make sure we have received all the events before releasing a window. For this, the Time-To-Postpone answer to the question, but only partially, as we do not allow to go further in past than the Time-To-Postpone.

A Pattern-based Approach for an Early Detection of Popular Twitter Accounts

Jonathan Debure
AIRBUS & CNAM, Paris, France
jonathan.debure@airbus.com

Stephan Brunessaux
AIRBUS, Paris, France
stephan.brunessaux@airbus.com

Camelia Constantin
Sorbonne University, Paris, France
camelia.constantin@lip6.fr

Cédric du Mouza
CNAM, Paris, France
dumouza@cnam.fr

ABSTRACT

Social networks (SN) are omnipresent in our lives today. Not all users have the same behaviour on these networks. If some have a low activity, rarely posting messages and following few users, some others at the other extreme have a significant activity, with many followers and regularly posts. The important role of these popular SN users makes them the target of many applications for example for content monitoring or advertising. It is therefore relevant to be able to predict as soon as possible which SN users will become popular.

In this work, we propose a technique for early detection of such users based on the identification of characteristic patterns. We present an index, H^2M, which allows a scaling up of our approach to large social networks. We also describe our first experiments that confirm the validity of our approach.

1 PROBLEM STATEMENT AND OUR CONTRIBUTION

Online social networks have become nowadays an essential means for communication, entertainment and marketing. Platforms like YouTube, Facebook, Twitter and Instagram gather hundreds of millions of users every day. While they have their own specifics and propose different content and interactions ways, these platforms share some common characteristics: first, their large number of users and the phenomenal amount of data (texts, pictures, videos, etc) produced daily; second, their network structure, with users connected to other users to share content; third, their high dynamicity with new users joining the platforms, others leaving, and connections between users which are continuously created or deleted.

These different characteristics make these platforms a tool particularly used to communicate information to a large number of people. In these networks, the most popular and influential users have quickly been the center of attention for many applications, since they will accelerate the spread of information to the greatest number of users [4]. For instance, for online advertising campaigns on social networks or on the Web, advertisers seek to place their advertisements among the users who have the most visibility in order to reach a maximum of people [1, 2, 5]. Likewise, for marketing purposes, highly followed users, called *influencers*, are paid to test and promote different products. In another area, popular users allow messages to be transmitted to a large audience for which social networks are the main media of information. These are the users who can quickly spread fake news or on the contrary bring a denial [3, 6]. Checking the content they publish is therefore particularly important. In the area of security, monitoring the content posted by some popular users who use social media for propaganda and / or indoctrination is also essential.

The various existing works offer techniques for detecting users who are already popular or influential in social networks. However, the various examples of applications presented above show that it is important to be able to identify the appearance of popular users on social networks as soon as possible. This article is, to our knowledge, the first to try to identify users who are on the way to more or less near future, to become popular. By detecting recurring patterns in the evolution of the popularity of accounts becoming popular, we manage with good precision to detect users several weeks before they become really popular. In addition, the index structure that we offer makes it possible to scale up to hundreds of millions of users and therefore allow our solution to be deployed for real social media platforms. Our experiences with real Twitter datasets validate our approach.

In summary, the contributions of our article are as follows:

(1) a characterization of the evolution of popularity for different classes of users (popular, non-popular, becoming popular);
(2) a pattern-based approach for early detection of popular users;
(3) an indexing structure for an efficient pattern-matching which scales to hundreds of millions of users to an early detection of future popular users;
(4) a validation on a large real Twitter dataset.

The rest of the paper is organized as follows. Section **??** reviews the related work. Section **??** describes the data model for popularity evolution. An analysis of a real Twitter dataset and the patterns we extracted for different classes of users is presented in Section **??**. We introduce our pattern-based approach along with its indexing structure for an early detection of users becoming popular in Section **??**. Section **??** gathers some of the experiments we perform to validate our approach. We conclude the paper and introduce some future work in Section **??**.

REFERENCES

[1] Zeinab Abbassi, Aditya Bhaskara, and Vishal Misra. Optimizing Display Advertising in Online Social Networks. In Aldo Gangemi, Stefano Leonardi, and Alessandro

Panconesi, editors, *Proc. Intl. Conf. on World Wide Web, (WWW) 2015*, pages 1–11. ACM, 2015.

[2] Çigdem Aslay, Wei Lu, Francesco Bonchi, Amit Goyal, and Laks V. S. Lakshmanan. Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret. *Proc. VLDB Endow.*, 8(7):822–833, 2015.

[3] Cody Buntain and Jennifer Golbeck. Automatically Identifying Fake News in Popular Twitter Threads. In *Proc. IEEE Intl. Conf. on Smart Cloud (SmartCloud)*, pages 208–215. IEEE Computer Society, 2017.

[4] Qi Cao, Huawei Shen, Jinhua Gao, Bingzheng Wei, and Xueqi Cheng. Popularity Prediction on Social Platforms with Coupled Graph Neural Networks. In *Proc. ACM Intl. Conf. on Web Search and Data Mining WSDM*, pages 70–78. ACM, 2020.

[5] David Dupuis, Cédric du Mouza, Nicolas Travers, and Gaël Chareyron. RTIM: A Real-Time Influence Maximization Strategy. In Reynold Cheng, Nikos Mamoulis, Yizhou Sun, and Xin Huang, editors, *Proc. Intl. Conf. on Web Information Systems Engineering (WISE)*.

[6] Zilong Zhao, Jichang Zhao, Yukie Sano, Orr Levy, Hideki Takayasu, Misako Takayasu, Daqing Li, Junjie Wu, and Shlomo Havlin. Fake News Propagates Differently from Real News even at Early Stages of spreading. *EPJ Data Sci.*, 9(1):7, 2020.

Cache-aware scheduling of scientific workflows in multisite cloud

Gaëtan Heidsieck
gaetan.heidsieck@inria.fr
Inria, University of Montpellier, CNRS, LIRMM
Montpellier, France

Daniel de Oliveira
University of Fluminence (UFF)
Niterói, Brazil

Esther Pacitti
LIRMM, University of Montpellier, CNRS, Inria
Montpellier, France

Christophe Pradal
CIRAD, AGAP
Montpellier, France

François Tardieu
INRAE, LEPSE
Montpellier, France

Patrick Valduriez
Inria, University of Montpellier, CNRS, LIRMM
Montpellier, France

ABSTRACT

In many scientific domains, *e.g*, bio-science [10], complex numerical experiments typically require many processing or analysis steps over huge datasets. They can be represented as scientific workflows, or workflows for short in this paper (but not to be confused with business workflows). These workflows facilitate the modeling, management, and execution of computational activities linked by data dependencies. As the size of the data processed and the complexity of the computation keep increasing, these workflows become data-intensive [10], thus requiring high-performance computing resources.

The cloud is a convenient infrastructure for handling workflows, as it allows leasing resources at a very large scale and relatively low cost. In this paper, we consider the execution of a data-intensive workflow in a multisite cloud, *i.e*., a cloud with geo-distributed data centers (henceforth named sites). Today, all popular public clouds, *e.g*, Microsoft Azure, Amazon EC2, and Google Cloud, provide a multisite option with the capability of using multiple sites with a single cloud account, The main reason for using multiple cloud sites to execute data-intensive workflows is that they often exceed the capabilities of a single site, either because the site imposes usage limits for fairness and security, or simply because the datasets are too large.

In scientific applications, there can be much heterogeneity in the storage and computing capabilities of the different sites, *e.g*, on premise servers, HPC platforms from research organizations or federated cloud sites at the national level [4]. As an example in plant phenotyping, greenhouse platforms generate terabytes of raw data from plants. Such data is typically stored at data centers geographically close to the greenhouse to minimize the impact of data transfers. However, the computation power of those data centers may be limited and fail to scale when the analyses become complex, such as in plant modeling or 3D reconstruction. In this case, other computation sites are then required.

Most scientific workflow management systems, or workflow systems for short, can execute workflows in the cloud [5]. Some examples are Swift/T, Pegasus, SciCumulus, Kepler and OpenAlea [6, 7, 11, 18, 20]. Our work is based on OpenAlea [18], which is widely used in plant science for simulation and analysis [17]. Most existing systems use naive or manual approaches to distribute the tasks across sites. The problem of scheduling a workflow execution over a multisite cloud has started to be addressed in [14], using performance models to predict the execution time on different resources. In [13], it is proposed a solution based on multi-objective scheduling and a single site virtual machine provisioning approach, assuming homogeneous sites, as in public cloud.

Since it is common for workflow users to reuse code or data from other workflows and from previous executions of the same workflow [8], a promising approach for efficient workflow execution is to cache intermediate data in order to avoid re-executing entire workflows. Furthermore, a user may need to re-execute a workflow many times with different sets of parameters and input data depending on the previous results. Workflow fragments, *i.e*., subsets of workflow activities and dependencies, can often be reused. Another important benefit of caching intermediate data is to make it easy to share with other research teams, thus fostering new analyses at low cost.

Caching has been supported by some workflow systems, *e.g*, Kepler [1], VisTrails [3] and OpenAlea [16]. In [9], we proposed an adaptive caching method for OpenAlea that automatically determines the most suited intermediate data to cache, taking into account workflow fragments, but only in the case of a single cloud site. Another interesting single site method, also exploiting workflow fragments, is to compute the ratio between re-computation cost and storage cost to determine what intermediate data should be stored [21]. All these methods are designed for a single site. The only distributed caching method for workflow execution in a multisite cloud we are aware of is restricted to hot metadata (frequently accessed metadata) and ignores intermediate data [12].

Caching data in a multisite cloud with heterogeneous sites is much more complex. In addition to the trade-off between re-computation and storage cost at single sites, there is the problem of site selection for placing cached data. The problem is more difficult than data allocation in distributed databases [15], which deals only with well-defined base data, not intermediate data produced. Furthermore, the scheduling of workflow executions must be cache-aware, *i.e*., exploit the knowledge of cached data to decide between reusing

and transferring cached data versus re-executing the workflow fragments.

The system design of our solution has been motivated by a real use case in plant phenotyping. It will also serve as a basis for the experimental evaluation. In the last decade, high-throughput phenotyping platforms have emerged, allowing for the acquisition of quantitative data on thousands of plants in well-controlled environmental conditions. For instance, the seven facilities of the French Phenome project [1] produce each year 200 Terabytes of data, which are various (images, environmental conditions and sensor outputs), multiscale and coming from different cloud sites. Analyzing such massive datasets is an open, yet important, problem for biologists [19].

The Phenomenal workflow [2], has been developed in OpenAlea to analyze and reconstruct the geometry and topology of thousands of plants through time in various conditions. Phenomenal is continuously evolving with the addition of new state-of-the-art methods, thus yielding new biological insights. The raw data is produced by the Phenoarch platform, which has a capacity of managing 1,680 plants within a controlled environment (*e.g*, temperature, humidity, irrigation) and automatic imaging through time. The total size of the raw image dataset for one experiment is 11 Terabytes.

In this paper, we propose a solution for cache-aware scheduling of scientific workflows in multisite cloud. Our solution is based on a distributed and parallel architecture and includes new algorithms for adaptive caching, cache site selection and dynamic workflow scheduling. Our architecture capitalizes on the latest advances in distributed and parallel data management to offer performance and scalability [15]. We consider a distributed cloud architecture with on premise servers, where raw data is produced, *e.g*, by a phenotyping experimental platform, and remote sites, where the workflow is executed. The remote sites are data centers using shared-nothing clusters, *i.e.*, clusters of server machines, each with independent processors, disk and memory. We adopt shared-nothing as it is the most scalable and cost-effective architecture for big data analysis.

We implemented our solution in the OpenAlea workflow system and performed an extensive experimental evaluation in a three-site cloud with a real application in plant phenotyping (Phenomenal). We compared our solution with two baselines: 1) a multisite workflow scheduling algorithm that does not consider intermediate data cache, 2) and a centralized cache architecture for workflow execution. For further comparisons, we extended two multisite scheduling algorithms to exploit our caching architecture. We showed that our solution for caching and reusing intermediate data can reduce the total workflow execution up to 42% with 60% of same input data for each new execution.

REFERENCES

[1] Ilkay Altintas, Oscar Barney, and Efrat Jaeger-Frank. 2006. Provenance collection support in the kepler scientific workflow system. In *International Provenance and Annotation Workshop*. 118–132.

[2] Simon Artzet, Nicolas Brichet, Jerome Chopard, Michael Mielewczik, Christian Fournier, and Christophe Pradal. 2018. OpenAlea.Phenomenal: A Workflow for Plant Phenotyping. https://doi.org/10.5281/zenodo.1436634

[3] Steven P Callahan, Juliana Freire, Emanuele Santos, Carlos E Scheidegger, Cláudio T Silva, and Huy T Vo. 2006. VisTrails: visualization meets data management. In *ACM SIGMOD Int. Conf. on Management of Data (SIGMOD)*. 745–747.

[4] Steve Crago, Kyle Dunn, Patrick Eads, Lorin Hochstein, Dong-In Kang, Mikyung Kang, Devendra Modium, Karandeep Singh, Jinwoo Suh, and John Paul Walters. 2011. Heterogeneous cloud computing. In *2011 IEEE International Conference on Cluster Computing*. IEEE, 378–385.

[5] Daniel de Oliveira, Fernanda Araujo Baião, and Marta Mattoso. 2010. Towards a taxonomy for cloud computing from an e-science perspective. In *Cloud Computing. Computer Communications and Networks*. Springer, 47–62.

[6] Daniel de Oliveira, Eduardo Ogasawara, Fernanda Baião, and Marta Mattoso. 2010. Scicumulus: A lightweight cloud middleware to explore many task computing paradigm in scientific workflows. In *2010 IEEE 3rd International Conference on Cloud Computing*. IEEE, 378–385.

[7] Ewa Deelman, Karan Vahi, Mats Rynge, Gideon Juve, Rajiv Mayani, and Rafael Ferreira Da Silva. 2016. Pegasus in the cloud: Science automation through workflow technologies. *IEEE Internet Computing* 20, 1 (2016), 70–76.

[8] Daniel Garijo, Pinar Alper, Khalid Belhajjame, Oscar Corcho, Yolanda Gil, and Carole Goble. 2014. Common motifs in scientific workflows: An empirical analysis. *Future Generation Computer Systems (FGCS)* 36 (2014), 338–351.

[9] Gaëtan Heidsieck, Daniel de Oliveira, Esther Pacitti, Christophe Pradal, François Tardieu, and Patrick Valduriez. 2019. Adaptive Caching for Data-Intensive Scientific Workflows in the Cloud. In *Int. Conf. on Database and Expert Systems Applications (DEXA)*. 452–466.

[10] Steve Kelling, Wesley M Hochachka, Daniel Fink, Mirek Riedewald, Rich Caruana, Grant Ballard, and Giles Hooker. 2009. Data-intensive science: a new paradigm for biodiversity studies. *BioScience* 59, 7 (2009), 613–620.

[11] Prakashan Korambath, Jianwu Wang, Ankur Kumar, Lorin Hochstein, Brian Schott, Robert Graybill, Michael Baldea, and Jim Davis. 2014. Deploying kepler workflows as services on a cloud infrastructure for smart manufacturing. *Procedia Computer Science* 29 (2014), 2254–2259.

[12] Ji Liu, Luis Pineda Morales, Esther Pacitti, Alexandru Costan, Patrick Valduriez, Gabriel Antoniu, and Marta Mattoso. 2018. Efficient Scheduling of Scientific Workflows using Hot Metadata in a Multisite Cloud. *IEEE Trans. on Knowledge and Data Engineering* (2018), 1–20.

[13] Ji Liu, Esther Pacitti, Patrick Valduriez, Daniel de Oliveira, and Marta Mattoso. 2016. Multi-objective scheduling of Scientific Workflows in multisite clouds. *Future Generation Computer Systems(FGCS)* 63 (2016), 76–95.

[14] Ketan Maheshwari, Eun-Sung Jung, Jiayuan Meng, Venkatram Vishwanath, and Rajkumar Kettimuthu. 2014. Improving Multisite Workflow Performance Using Model-Based Scheduling. In *IEEE nt. Conf. on Parallel Processing (ICPP)*. 131–140.

[15] M. Tamer Özsu and Patrick Valduriez. 2020. *Principles of Distributed Database Systems, Fourth Edition.* Springer.

[16] Christophe Pradal, Simon Artzet, Jerome Chopard, Dimitri Dupuis, Christian Fournier, Michael Mielewczik, Vincent Negre, Pascal Neveu, Didier Parigot, Patrick Valduriez, et al. 2017. InfraPhenoGrid: a scientific workflow infrastructure for plant phenomics on the grid. *Future Generation Computer Systems (FGCS)* 67 (2017), 341–353.

[17] Christophe Pradal, Samuel Dufour-Kowalski, Frédéric Boudon, Christian Fournier, and Christophe Godin. 2008. OpenAlea: a visual programming and component-based software platform for plant modelling. *Functional plant biology* 35, 10 (2008), 751–760.

[18] Christophe Pradal, Christian Fournier, Patrick Valduriez, and Sarah Cohen-Boulakia. 2015. OpenAlea: scientific workflows combining data analysis and simulation. In *Int. Conf. on Scientific and Statistical Database Management (SSDBM)*. 11:1–11:6.

[19] François Tardieu, Llorenç Cabrera-Bosquet, Tony Pridmore, and Malcolm Bennett. 2017. Plant phenomics, from sensors to knowledge. *Current Biology* 27, 15 (2017), R770–R783.

[20] Justin M Wozniak, Timothy G Armstrong, Michael Wilde, Daniel S Katz, Ewing Lusk, and Ian T Foster. 2013. Swift/t: Large-scale application composition via distributed-memory dataflow processing. In *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. IEEE, 95–102.

[21] Dong Yuan, Yun Yang, Xiao Liu, Wenhao Li, Lizhen Cui, Meng Xu, and Jinjun Chen. 2013. A highly practical approach toward achieving minimum data sets storage cost in the cloud. *IEEE Trans. on Parallel and Distributed Systems* 24, 6 (2013), 1234–1244.

[1] https://www.phenome-emphasis.fr/phenome_eng/

Efficient Discovery of Compact Maximal Behavioral Patterns from Event Logs

Mehdi Acheli
Daniela Grigori
mehdi.acheli@dauphine.fr
daniela.grigori@dauphine.fr
Univ. Paris-Dauphine, CNRS UMR[7243], LAMSADE,
75016 Paris, France

Matthias Weidlich
matthias.weidlich@hu-berlin.de
Humboldt-Universität zu Berlin, Germany

ABSTRACT

Techniques for process discovery support the analysis of information systems by constructing process models from event logs that are recorded during system execution. In recent years, various algorithms to discover end-to-end process models have been proposed. Yet, they do not cater for domains in which process execution is highly flexible, as the unstructuredness of the resulting models renders them meaningless. It has therefore been suggested to derive insights about flexible processes by mining behavioral patterns, i.e., models of frequently recurring episodes of a process' behavior. However, existing algorithms to mine such patterns suffer from imprecision and redundancy of the mined patterns and a comparatively high computational effort. In this work, we overcome these limitations with a novel algorithm, coined COBPAM (COmbination based Behavioral PAttern Mining). It exploits a partial order on potential patterns to discover only those that are compact and maximal, i.e. least redundant. Moreover, COBPAM exploits that complex patterns can be characterized as combinations of simpler patterns, which enables pruning of the pattern search space. Efficiency is improved further by evaluating potential patterns solely on parts of an event log. Experiments with real-world data demonstrates how COBPAM improves over the state-of-the-art in behavioral pattern mining.

KEYWORDS

Behavioral Patterns, Process Discovery, Pattern Mining

Algorithmes à base de provenance pour des requêtes enrichies sur les bases de données graphes

Yann Ramusat
DI ENS, ENS, CNRS,
Université PSL & Inria
Paris, France
yann.ramusat@ens.fr

Silviu Maniu
Université Paris-Saclay, LRI, CNRS
& Inria
Paris, France
silviu.maniu@lri.fr

Pierre Senellart
DI ENS, ENS, CNRS,
Université PSL & Inria & IUF
Paris, France
pierre@senellart.com

Les bases de données orientées graphe [13] font partie de l'écosystème des SGBD appelés NoSQL, dans lesquels l'information n'est pas organisée en suivant strictement le modèle relationnel. La structure des bases de données graphe est bien adaptée à la représentation de certains types de relations dans les données et leur potentiel pour la distribution les rendent attractives pour des applications nécessitant du stockage à grande échelle et du traitement de données massivement parallèle. Des exemples d'applications naturelles de tels systèmes de bases de données sont l'analyse des réseaux sociaux [5] ou le stockage et l'interrogation du Web sémantique [2].

Les bases de données graphe peuvent être interrogées en utilisant plusieurs langages de requêtes généraux de navigation, dont une abstraction est les *requêtes régulières de chemin* (*regular path queries* ou *RPQ* en anglais) [3] (ou des généralisations de celles-ci, comme les C2RPQ), sur les chemins du graphe. Récemment, en s'appuyant sur les solutions existantes pour l'interrogation des graphes à propriétés – comme le langage Cypher [6] de Neo4j ou PGQL [15] d'Oracle – une future norme internationale pour l'interrogation de graphes à propriétés, GQL [9], est en cours d'élaboration en tant que langage de requête à part entière au côté de SQL. GQL inclura notamment un support des RPQ.

En parallèle de ces développements récents, la notion de *provenance* d'un résultat de requête [14], une notion familière dans les bases de données relationnelles, a récemment été adaptée au contexte des bases de données graphe [11], en utilisant le cadre des semi-anneaux de provenance [7]. Dans ce cadre, les arêtes d'un graphe sont annotées, en plus des propriétés usuelles, par des éléments d'un semi-anneau ; quand une requête est évaluée, le fait de traverser les chemins du graphe peut engendrer de nouvelles annotations qui dépendent des opérateurs du semi-anneau, et qui résultent en une valeur du semi-anneau associée à chaque résultat de la requête, appelée la provenance du résultat. En choisissant différents semi-anneaux, des informations différentes sur le résultat de la requête peuvent être calculées. Par exemple, quand les arêtes sont annotées avec des éléments du semi-anneau *tropical* (les nombres réels positifs ou nuls) exprimant la distance entre les nœuds, la provenance du résultat calcule la plus courte distance des chemins qui ont produit ce résultat ; quand les arêtes sont annotées par des éléments du semi-anneau de *comptage* (les entiers naturels) interprétés comme une multiplicité, la provenance du résultat calcule le nombre (qui peut être infini en cas de cycles) de manière dont chaque résultat peut être obtenu. Les propriétés sous-jacentes du

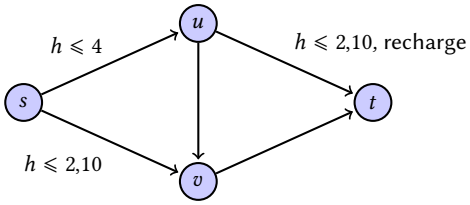


Figure 1: Réseau routier exemple représenté par un graphe avec des annotations de provenance selon deux dimensions : la hauteur h maximale (un nombre positif) qu'un véhicule doit avoir pour utiliser le segment de route, et un booléen indiquant la présence d'une station de recharge pour véhicule électrique. Quand une dimension n'est pas mentionnée, les annotations sont supposées être, respectivement, $h \leqslant \infty$ et $\neg(\text{recharge})$.

semi-anneau contrôlent directement la manière dont l'information sur les arêtes du graphe est encodée et également l'efficacité des algorithmes de traitement des requêtes.

Au-delà de ces exemples simples de semi-anneaux, le cadre de la provenance par semi-anneau permet aussi de modéliser des problèmes complexes, p. ex., où le problème d'intérêt peux être décomposé en plusieurs sous-problèmes et où la provenance du résultat ne correspond pas nécessairement à un chemin particulier dans le graphe.

Exemple 1. *Considérons l'exemple d'un réseau de transport routier modélisé comme un graphe orienté avec des annotations de provenance sur les arêtes. On peut par exemple encoder la présence de points d'intérêts (tels que des stations essence, des restaurants ou des stations de recharge électrique) comme des caractéristiques booléennes des arêtes, et les propriétés des routes (p. ex., hauteur ou poids maximal pour un tunnel ou un pont) comme des caractéristiques à valeur réelle.*

Nous allons montrer que, en utilisant la provenance par semi-anneaux, nous pouvons traiter des requêtes de graphe qui prennent en compte une multiplicité de telles caractéristiques : une paire de nœuds est valide pour ces requêtes s'il existe au moins un chemin valide pour chaque restriction entre les deux emplacements. Une application de cela serait de s'assurer que différentes catégories de véhicules (disons, un camion de grand gabarit et une voiture électrique nécessitant une recharge sur le chemin) peuvent atteindre une destination commune à partir de la même origine.

Une autre sémantique possible pour la provenance par semi-anneaux est de vérifier que tous les chemins entre deux nœuds vérifient (ou excluent) certaines propriétés (p. ex., absence de péages ou présence de

stations essence sur la route) fournissant ainsi à des administrateurs des informations cruciales sur l'état global des itinéraires entre deux points.

Ceci est illustré en figure 1, un réseau routier dans lequel certains segments de routes ont des restrictions sur la hauteur des véhicules ; c'est une première dimension de provenance. La deuxième dimension indique s'il existe une station de recharge électrique sur le segment de route – dans notre exemple, ce n'est le cas que pour une seule arête.

Dans nos recherches préliminaires antérieures [11], nous avons généralisé trois algorithmes existants d'une large gamme de la littérature en informatique au calcul de la provenance de requêtes régulières de chemin, dans le cadre de provenance par semi-anneaux. Pris ensemble, ces trois généralisations recouvrent une grande classe de semi-anneaux utilisés pour la provenance, chacun conduisant à un compromis entre complexité en temps et généralité. Nous avons également conduit des expériences suggérant que ces approches sont complémentaires et applicables en pratique pour divers types d'indications de provenance, même sur des réseaux de transports relativement grands.

Dans les recherches résumées ici, et décrites en détail en [12], nous étendons ce travail en :

- Introduisant un nouvel algorithme, MultiDijkstra, pour les semi-anneaux commutatifs *0-clos* (ou *absorptifs*). Cet algorithme, qui généralise l'algorithme de Dijkstra et exploite les propriétés des treillis distributifs, comble partiellement un fossé entre deux classes de semi-anneaux qui était non traité dans nos recherches antérieures. Les requêtes de l'exemple 1 font partie de cette classe et ont fortement motivé notre intérêt pour développer de nouveaux algorithmes. Les expériences que nous avons conduites démontrent que notre nouvel algorithme passe à l'échelle de très grands réseaux contenant des dizaines de millions de nœuds, apportant une amélioration notable à l'état de l'art du calcul de provenance dans les bases de données graphe.
- Établissant un résumé précis, sous la forme d'une taxonomie, des algorithmes utilisés dans notre contexte, ainsi que de leur complexité et des propriétés attendues des semi-anneaux sous-jacents utilisés pour les annotations de provenance. Nous analysons également les similarités avec des classes de semi-anneaux utilisés soit pour le calcul de provenance de requêtes de l'algèbre relationnelle [8] soit pour celui de programmes Datalog [4].
- Accomplissant un ensemble complet d'expériences sur des données du monde réel démontrant le temps de calcul de la provenance sur des graphes, avec une grande de variété de semi-anneaux et de cas d'utilisations. Nous observons également que les paramètres de topologie du graphe, comme la *largeur d'arbre* [10] semblent avoir un impact plus important sur l'efficacité des algorithmes que des paramètres basés sur la distance tels que la *highway dimension* [1]. L'implémentation de tous les algorithmes que nous utilisons pour ces expériences est librement disponible sur https://bitbucket.org/smaniu/graph-provenance/src/master/.

Pour plus de détails sur ce travail, se référer à [12].

REMERCIEMENTS

Ce travail a été financé par le gouvernement français sous gestion de l'Agence Nationale de la Recherche comme partie du programme « Investissements d'avenir », référence ANR-19-P3IA-0001 (Institut 3IA PRAIRIE).

RÉFÉRENCES

[1] Ittai Abraham, Amos Fiat, Andrew V. Goldberg, and Renato Fonseca F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *SODA*, pages 782–793, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[2] Marcelo Arenas and Jorge Pérez. Querying semantic web data with sparql. In *PODS*, pages 305–316, New York, 2011.

[3] Pablo Barceló. Querying graph databases. In *PODS*, pages 175–188, New York, 2013. ACM.

[4] Daniel Deutch, Tova Milo, Sudeepa Roy, and Val Tannen. Circuits for Datalog Provenance. In *ICDT*, pages 201–212, 2014.

[5] Pedro Domingos and Matthew Richardson. Mining the network value of customers. In *KDD*, pages 57–66, New York, 2001. ACM.

[6] Nadime Francis, Andrés Taylor, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, and Petra Selmer. Cypher : An evolving query language for property graphs. In *SIGMOD*, pages 1433–1445, 2018.

[7] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *PODS*, pages 31–40, New York, 2007. ACM.

[8] Todd J. Green and Val Tannen. The semiring framework for database provenance. In *PODS*, page 93–99, New York, NY, USA, 2017. Association for Computing Machinery.

[9] ISO SC32 / WG3. Graph Query Language GQL. https://www.gqlstandards.org/.

[10] Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data. In *ICDT*, Lisbon, Portugal, 2019.

[11] Yann Ramusat, Silviu Maniu, and Pierre Senellart. Semiring provenance over graph databases. In *TaPP*, 2018.

[12] Yann Ramusat, Silviu Maniu, and Pierre Senellart. Provenance-based algorithms for rich queries over graph databases, 2021. À paraître.

[13] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph Databases*. O'Reilly Media, 2013.

[14] Pierre Senellart. Provenance and probabilities in relational databases: From theory to practice. *SIGMOD Record*, 46(4), 2017.

[15] Oskar van Rest, Sungpack Hong, Jinha Kim, Xuming Meng, and Hassan Chafi. PGQL : A property graph query language. In *GRADES*, pages 7 :1–7 :6, New York, NY, USA, 2016. ACM.

Traitement coopératif du problème des réponses pléthoriques dans les bases de connaissances RDF

Louise Parkin
louise.parkin@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

Ibrahim Dellal
ibrahim.dellal@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

Brice Chardin
brice.chardin@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

Stéphane Jean
stephane.jean@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

Allel Hadjali
allel.hadjali@ensma.fr
LIAS, ISAE-ENSMA
Chasseneuil-du-Poitou, France

RÉSUMÉ

Les utilisateurs d'une base de connaissances sont confrontés à un grand volume de données dont ils peuvent ignorer la structure sous-jacente. Ainsi, ils peuvent commettre des erreurs dans la formulation de leurs requêtes et obtenir des réponses non satisfaisantes. Nous nous intéressons ici au cas particulier du problème des réponses pléthoriques, où une requête produit beaucoup plus de résultats que n'attendait l'utilisateur. L'approche la plus connue pour traiter ce problème, la méthode dite top-K, consiste à classer les résultats pour ne retourner que les meilleures réponses. Cependant, si la requête comporte de mauvaises préconceptions, cette stratégie ne règle pas la source du problème et donc ne constitue pas une solution satisfaisante. Nous proposons donc une nouvelle méthode coopérative, permettant aux utilisateurs de comprendre l'origine des réponses pléthoriques de leur requête. Pour cela nous fournissons deux informations : (i) les parties minimales de la requête entraînant des réponses pléthoriques, et (ii) les parties maximales de la requête dont les réponses ne sont pas pléthoriques. Pour identifier ces deux informations, nous proposons deux algorithmes et montrons leur efficacité par rapport à une méthode naïve en utilisant des données synthétiques et réelles.

1 INTRODUCTION

Le développement du Web sémantique a permis la création de multiples bases de connaissances (BC) académiques et industrielles. Les BC stockent l'information sous forme de triplets RDF (sujet, prédicat, objet) et sont interrogées via le langage SPARQL [1]. Un utilisateur d'une BC a rarement une connaissance approfondie des données manipulées, ce qui peut le conduire à formuler des requêtes basées sur des préconceptions erronées et donnant ainsi des réponses insatisfaisantes. Les différents types de réponses insatisfaisantes sont : les réponses vides, les réponses pléthoriques, les réponses insuffisantes, l'absence d'une réponse attendue ou la présence d'une réponse inattendue. Nous nous intéressons ici au deuxième problème, où l'utilisateur reçoit un nombre trop important de réponses et ne peut pas en extraire des informations pertinentes. On parle de réponses pléthoriques lorsque le nombre de réponses dépasse un seuil K prédéfini.

L'approche la plus connue pour ce problème, la méthode top-K, repose sur un classement des réponses et une sélection des K premiers résultats. Si cette stratégie limite le nombre de réponses à K, elle n'agit pas au niveau de la requête, et ne peut donc pas traiter les problèmes que celle-ci peut présenter. Aussi, nous nous basons sur les solutions proposées dans le cadre du problème des réponses vides [2] pour fournir deux notions coopératives qui aideront les utilisateurs à comprendre l'origine des réponses pléthoriques. Nous fournissons d'abord des causes d'échec, appelées MFIS (*Minimal Failure Inducing Subqueries*). Ce sont les plus petites parties de la requête qui produisent des réponses pléthoriques sur lesquelles l'utilisateur devra focaliser son attention afin de modifier sa requête. Nous fournissons également des requêtes alternatives, appelées XSS (*maXimal Succeeding Subqueries*). Ce sont les plus grandes parties de la requête qui produisent des réponses non pléthoriques. Elles peuvent être utilisées à la place de la requête originale avec une garantie qu'un nombre de réponses n'excédant pas K sera retourné. A partir d'un algorithme naïf nécessitant l'exécution d'un nombre exponentiel de requêtes, nous proposons trois améliorations qui exploitent des propriétés identifiées sur les requêtes ou les données manipulées. Nous montrons leur intérêt expérimentalement en utilisant des données et requêtes générées avec le banc d'essai WatDiv [3], ainsi que des données et requêtes réelles provenant respectivement de DBpedia et du projet Linked SPARQL Queries Dataset [4].

2 ÉTAT DE L'ART

Les approches existantes pour le problème des réponses pléthoriques sont de deux types : les approches orientées données et les approches orientées requêtes. Les méthodes orientées données, telles que les approches top-K [5] ou les stratégies de groupement [6, 7] considèrent que la requête est correcte et cherchent à présenter les résultats de façon synthétique. Comme les méthodes orientées données ne s'intéressent pas à la requête, elles ne sont pas adaptées dans les situations où la requête pose problème.

A l'inverse, les méthodes orientées requêtes considèrent que la requête est à l'origine des réponses pléthoriques et proposent de la modifier. Cette modification peut se faire en ajoutant [8], supprimant [9] ou modifiant les patrons de triplets présents dans la requête [10, 11]. Cependant, les solutions existantes le font sans

chercher à identifier les causes des réponses pléthoriques. C'est l'objet de nos travaux.

3 DÉFINITION ET CALCUL DES MFIS ET XSS

Les sous-requêtes d'une requête conjonctive $Q = t_1 \wedge \cdots \wedge t_n$ sont les requêtes $Q' = t_i \wedge \cdots \wedge t_j$ où $\{t_i, \cdots, t_j\} \subseteq \{t_1, \cdots, t_n\}$, on dit alors que Q est une super-requête de Q'. Dans le cadre des réponses pléthoriques, une requête échoue si elle produit plus que K réponses, dans le cas contraire elle réussit. On appelle FIS (*Failure Inducing Subquery*) une sous-requête de la requête initiale qui échoue et dont toutes les super-requêtes échouent. Les MFIS (*Minimal FIS*) sont alors les FIS dont aucune sous-requête n'est une FIS. Les XSS (*maXimal Succeeding Subqueries*) sont les requêtes qui réussissent dont toutes les super-requêtes échouent.

Pour identifier les MFIS et XSS, une méthode naïve, appelée Base consiste à exécuter l'ensemble des sous-requêtes de la requête initiale. Pour une requête avec n patrons de triplets, cela implique d'exécuter $2^n - 1$ sous-requêtes. Pour réduire le temps de calcul des MFIS et XSS, nous introduisons des propriétés permettant de diminuer le nombre de requêtes à exécuter. Des définitions de MFIS et XSS, on déduit qu'une sous-requête d'une requête qui réussit ne peut être ni MFIS ni XSS, donc nous n'avons pas besoin de les exécuter. Une première amélioration, Bfs, exploite cette propriété.

La deuxième propriété que nous proposons affirme que si on retire à une requête $Q \wedge t$ un patron de triplet t en conservant toutes les variables, alors la nouvelle requête Q ne peut pas avoir moins de réponses que $Q \wedge t$. Ainsi, si $Q \wedge t$ échoue, on en déduit l'échec de Q sans l'exécuter. L'ajout de cette propriété constitue un nouvel algorithme, Var.

La dernière amélioration exploite une propriété qui implique à la fois les requêtes et les données. Nous introduisons le concept de cardinalité maximale d'un prédicat [12], qui décrit le nombre maximal d'occurrences d'un prédicat pour tous les sujets de la BC. La propriété associée indique que si on retire à une requête $Q \wedge t$ un patron de triplet t dont le prédicat a une cardinalité maximale de 1, alors la nouvelle requête Q ne peut pas avoir moins de réponses que $Q \wedge t$. Ainsi, si $Q \wedge t$ échoue, on en déduit l'échec de Q sans l'exécuter. En ajoutant cette propriété aux précédentes, on obtient notre dernier algorithme, Full.

Si les deux premières propriétés, et donc l'algorithme Var, peuvent s'appliquer à toutes les requêtes, la dernière nécessite de disposer d'une information supplémentaire : les cardinalités des prédicats de la BC. Comme il peut être difficile de maintenir de telles informations sur des bases de connaissances régulièrement modifiées, l'algorithme Full ne sera pas toujours applicable.

4 RÉSULTATS EXPÉRIMENTAUX

Nous avons évalué la performance de nos quatre algorithmes avec deux expérimentations réalisées sur le triplestore JenaTDB, en fixant le seuil des réponses pléthoriques à K=100. La première expérimentation utilise des données (11M triplets) et requêtes générées avec le banc d'essai WatDiv. Nous avons utilisé 21 requêtes, de forme étoile, chaîne et composite contenant 4 à 12 patrons de triplet. Les temps d'exécution mesurés pour les requêtes en étoile sont fournis en figure 1, avec le nombre de requêtes exécutées par chaque algorithme en table 1. On observe la même allure pour les autres formes de requêtes. Les algorithmes Bfs, Var et Full économisent respectivement 2%, 65% et 83% du temps d'exécution de l'algorithme naïf Base. On peut noter que même si l'algorithme Bfs peut exécuter la moitié du nombre de requêtes de Base, leurs temps d'exécution restent proches.

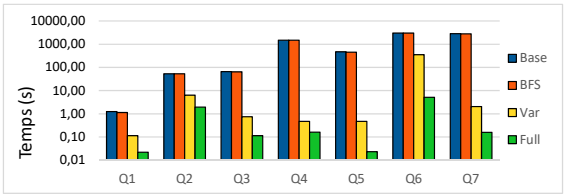


Figure 1: Temps d'exécution (requêtes en étoile - WatDiv)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Base	15	31	63	127	255	511	1023
BFS	15	31	33	75	129	511	513
Var	8	16	9	39	65	255	257
Full	2	4	2	12	3	8	9

Table 1: Nombre de requêtes exécutées (requêtes en étoile - WatDiv)

La seconde expérimentation se base sur une BC réelle, DBpedia, et des logs de requêtes réelles. Le comportement est proche de WatDiv, les algorithmes Bfs, Var et Full économisent respectivement 14%, 78% et 78% du temps d'exécution de l'algorithme Base. Dans ce cas, Full n'est pas significativement plus rapide que Var. Cela s'explique par les cardinalités de DBpedia qui permettent rarement d'exploiter la propriété supplémentaire de Full. Notre expérimentation montre que l'algorithme Var permet d'obtenir les MFIS et XSS dans un temps raisonnable, et que, selon les données manipulées, l'exploitation des cardinalités peut offrir une amélioration supplémentaire de performance.

RÉFÉRENCES

[1] Harris, S., Seaborne, A. : Sparql 1.1 query language. W3C Recommendation (2013)
[2] Godfrey, P. : Minimization in Cooperative Response to Failing Database Queries. International Journal of Cooperative Information Systems **6**(2) (1997) 95–149
[3] Aluç, G., Hartig, O., Özsu, M.T., Daudjee, K. : Diversified stress testing of rdf data management systems. In : ISWC'14, Springer (2014) 197–212
[4] Saleem, M., Ali, M.I., Hogan, A., Mehmood, Q., Ngomo, A.N. : LSQ : The Linked SPARQL Queries Dataset. In : ISWC'15. (2015) 261–269
[5] Ilyas, I.F., Beskales, G., Soliman, M.A. : A survey of top-k query processing techniques in relational database systems. ACM Comput. Surv. **40**(4) (2008)
[6] Chakrabarti, K., Chaudhuri, S., Hwang, S.w. : Automatic categorization of query results. In : SIGMOD'04. (2004) 755–766
[7] Ozawa, J., Yamada, K. : Discovery of global knowledge in a database for cooperative answering. In : IEEE'95. Volume 2. (1995) 849–854
[8] Bosc, P., Hadjali, A., Pivert, O., Smits, G. : Une approche fondée sur la corrélation entre prédicats pour le traitement des réponses pléthoriques. In : EGC'10. 273–284
[9] Vasilyeva, E., Thiele, M., Bornhövd, C., Lehner, W. : Answering "why empty ?" and "why so many ?" queries in graph databases. Journal of Computer and System Sciences **82**(1) (2016) 3–22
[10] Bosc, P., HadjAli, A., Pivert, O. : About overabundant answers to flexible queries. In : IPMU'06. Volume 6. (2006) 2221–2228
[11] Moises, S.A., Pereira, S.d.L. : Dealing with empty and overabundant answers to flexible queries. Journal of Data Analysis and Inf. Proc. (2014) 12–18
[12] Dellal, I. : Management and Exploitation of Large and Uncertain Knowledge Bases. PhD thesis, ISAE-ENSMA - Poitiers (2019)

A Partitioning Approach for Skyline Queries in Presence of Partial and Dynamic Orders

Karim Alami
Univ. Bordeaux, CNRS, LaBRI, UMR 5800
Talence, France
karim.alami@u-bordeaux.fr

Sofian Maabout
Univ. Bordeaux, CNRS, LaBRI, UMR 5800
Talence, France
sofian.maabout@u-bordeaux.fr

ABSTRACT

Consider the case of tourists looking for flight tickets. While one may assume that every tourist does prefer lower price, the preference among the airline companies is on the one hand *partial*, i.e., some companies may be incomparable, and on the other hand, it is *dynamic* in the sense that users have different preferences among the companies. In this paper, we address the problem of answering skyline queries in the presence of such partially and dynamically ordered attributes. The main idea of our solution consists in decomposing each query into a set of independent sub-queries with respect to the user's preference. Our contribution is twofold: (i) we propose an algorithm exploiting the above property to evaluate skyline queries on the fly and (ii) a pre-materialization of some sub-queries in order to optimize a query workload. We demonstrate empirically the efficiency of our proposals regarding its direct competitors.

KEYWORDS

Skyline queries, partial and dynamic order, data partitioning, materialization

Une dichotomie sur l'évaluation de requêtes closes sous homomorphismes sur les graphes probabilistes

Antoine Amarilli
LTCI, Télécom Paris, Institut Polytechnique de Paris

İsmail İlkan Ceylan
University of Oxford

ABSTRACT

Nous étudions le problème de l'évaluation probabiliste de requêtes (PQE) sur les graphes probabilistes, formalisés comme des bases de données probabilistes à tuples indépendants (TIDs) sur des signatures d'arité deux. Nous nous concentrons sur la classe des requêtes closes sous homomorphismes, que l'on peut également caractériser comme les unions infinies de requêtes conjonctives, et que nous dénotons donc par UCQ^∞. Notre résultat principal est de démontrer que le problème PQE est #P-difficile pour toutes les requêtes *non bornées* dans UCQ^∞. Comme les requêtes *bornées* de UCQ^∞ sont déjà classifiées par la dichotomie de Dalvi et Suciu [3], nos résultats et les leurs impliquent une dichotomie sur PQE portant sur l'ensemble des requêtes UCQ^∞ sur des graphes probabilistes. Cette dichotomie couvre en particulier tous les fragments de UCQ^∞ comme les requêtes *Datalog* (potentiellement disjonctives, mais sans négation), les *requêtes régulières de chemin* (RPQ), et une large classe de *requêtes exprimées à travers une ontologie* sur les signatures d'arité deux. Notre résultat est établi par une réduction depuis le problème de compter les valuations qui satisfont une formule 2-DNF positive partitionnée (#PP2DNF) pour certaines requêtes, ou depuis le problème de la probabilité de connexité entre une source et une cible dans un graphe non-orienté (#U-ST-CON) pour d'autres requêtes, en fonction des propriétés des modèles minimaux.

Cet article est une resoumission de notre travail présenté à la conférence ICDT'20 [1], qui a reçu le prix du meilleur article.

La version étendue contenant les preuves est disponible en ligne [2] à l'URL suivant : https://arxiv.org/abs/1910.02048.

REFERENCES

[1] Antoine Amarilli and İsmail İlkan Ceylan. A Dichotomy for Homomorphism-Closed Queries on Probabilistic Graphs. In *ICDT*, 2020.
[2] Antoine Amarilli and İsmail İlkan Ceylan. A Dichotomy for homomorphism-closed queries on probabilistic graphs, 2020. Full version with proofs: https://arxiv.org/abs/1910.02048.
[3] Nilesh Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *J. ACM*, 59(6), 2012.

Subsequence Anomaly Detection with Series2Graph

Paul Boniol
EDF R&D, LIPADE, Université de Paris
France
paul.boniol@edf.fr

Themis Palpanas
LIPADE, Université de Paris & French University Institute (IUF)
France
themis@mi.parisdescartes.fr

ABSTRACT

Subsequence anomaly detection in long sequences is an important problem with applications in a wide range of domains. However, the approaches that have been proposed so far in the literature have severe limitations: they either require prior domain knowledge that is used to design the anomaly discovery algorithms, or become cumbersome and expensive to use in situations with recurrent anomalies of the same type. In this work[1], we address these problems, and propose an unsupervised method suitable for subsequence anomaly detection. Our method, Series2Graph, is based on a graph representation of a novel low-dimensionality embedding of subsequences. Series2Graph needs neither labeled instances (like supervised techniques), nor anomaly-free data (like zero-positive learning techniques), and identifies anomalies of varying lengths. The experimental results, on the largest set of synthetic and real datasets used to date, demonstrate that the proposed approach correctly identifies single and recurrent anomalies without any prior knowledge of their characteristics, outperforming by a large margin several competing approaches in accuracy, while being up to orders of magnitude faster.

KEYWORDS

Time series, Data series, Subsequence anomalies, Outliers.

1 INTRODUCTION

Data series[2] anomaly detection is a crucial problem with application in a wide range of domains [3, 10]. Examples of such applications can be found in manufacturing, astronomy, engineering, and other domains, including detection of abnormal heartbeats in cardiology [8], wear and tear in bearings of rotating machines [2], machine degradation in manufacturing [9], hardware and software failures in data center monitoring [11], mechanical faults in vehicle operation monitoring [7] and identification of transient noise in gravitational wave detectors [4]. This implies a real need by relevant applications for developing methods that can accurately and efficiently achieve this goal.

[1]Originally published in PVLDB 13(11) 2020 [5, 6].
[2]A data series is an ordered sequence of real-valued points. If the dimension that imposes the ordering of the sequence is time then we talk about *time series*, but it could also be mass (e.g., mass spectrometry), angle (e.g., astronomy), or position (e.g., biology). In this paper, we will use the terms *time series*, *data series*, and *sequence* interchangeably.

[Anomaly Detection in Sequences] Anomaly detection is a well studied task that can be tackled by either examining single values, or sequences of points. In the specific context of sequences, which is the focus of this paper, we are interested in identifying anomalous subsequences [12, 15], which are not single abnormal values, but rather an abnormal *sequence* of values. In real-world applications, this distinction becomes crucial: in certain cases, even though each individual point may be normal, the trend exhibited by the sequence of these same values may be anomalous. Failing to identify such situations could lead to severe problems that may only be detected when it is too late.

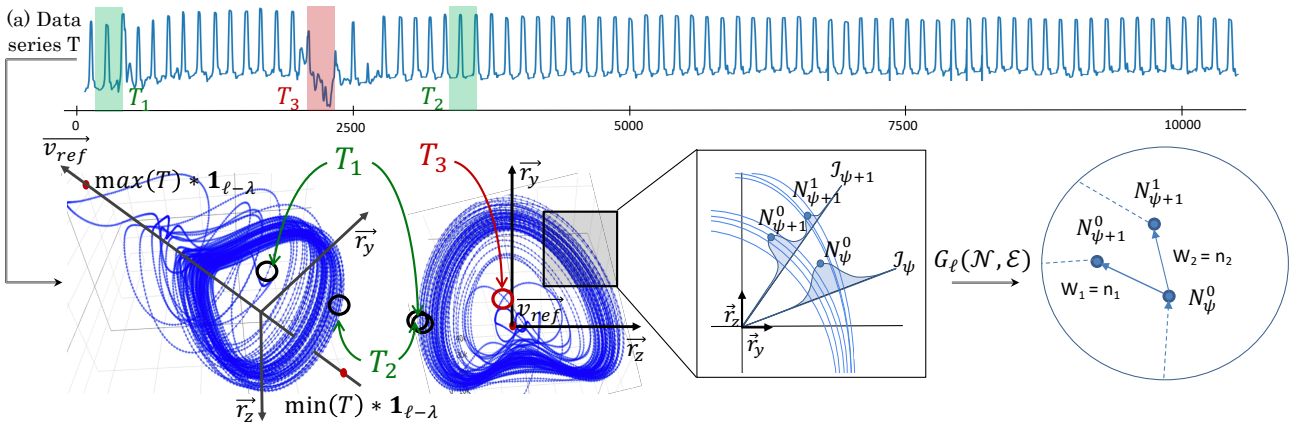
[Limitations of Previous Approaches] Some existing techniques explicitly look for a set of pre-determined types of anomalies [1, 8]. These are techniques that have been specifically designed to operate in a particular setting, they require domain expertise, and cannot generalize.

Other techniques identify as anomalies the subsequences with the largest distances to their nearest neighbors (termed discords) [12, 15]. The assumption is that the most distant subsequence is completely isolated from the "normal" subsequences. However, this definition fails when an anomaly repeats itself (approximately the same) [13]. In this situation, anomalies will have other anomalies as close neighbors, and will not be identified as discords. In order to remedy this situation, the $m^{th} discord$ approach has been proposed [14], which takes into account the multiplicity m of the anomalous subsequences that are similar to one another, and marks as anomalies all the subsequences in the same group. However, this approach assumes the cardinality of the anomalies to be known, which is not true in practice (otherwise, we need to try several different m values, increasing execution time). Furthermore, the majority of the previous approaches require prior knowledge of the anomaly length, and their performance deteriorates significantly when the correct length value is not used.

[Proposed Approach] In this work, we address the aforementioned issues, and we propose Series2Graph, an unsupervised method suitable subsequence anomaly detection. Our approach does not need labeled instances (like supervised techniques do), or clean data that do not contain anomalies (like zero-positive learning techniques require). It also allows the same model to be used for the detection of anomalies of different lengths.

Series2Graph is based on a graph representation of a novel low-dimensionality embedding of subsequences. Figure 1. depicts the different step involved to compute the resulting graph. These steps are the following:

- **Subsequence Embedding**: Project all the subsequences (of a given length ℓ) of T in a two-dimensional space, where shape similarity is preserved.



(b) First Step: Subsequence embedding (c) Second Step: Node creation (d) Third Step: Edge creation

Figure 1: Series2Graph steps in order to build the graph from a data series (a): embed the subsequences (b), create the nodes (c), and extract the edges (d).

- **Node Creation**: Create a node for each one of the densest parts of the above two-dimensional space. These nodes can be seen as a summarization of all the major patterns of length ℓ that occurred in T.
- **Edge Creation**: Retrieve all transitions between pairs of subsequences represented by two different nodes: each transition corresponds to a pair of subsequences, where one occurs immediately after the other in the input data series T. We represent transitions with an edge between the corresponding nodes. The weights of the edges are set to the number of times the corresponding pair of subsequences was observed in T.
- **Subsequence Scoring**: Compute the normality (or anomaly) score of a subsequence of length $\ell_q \geq \ell$ (within or outside of T), based on the previously computed edges/nodes and their weights/degrees.

This allows us then to differentiate between normal behavior, i.e., frequently occurring patterns, and anomalies, i.e., subsequences that rarely occur in the data series. Overall, the experimental results demonstrate that Series2Graph dominates by a large margin the competitors in accuracy, versatility, and execution time.

[Contributions] The aforementioned steps and details can be found in the original paper [6]. As a summary, our contributions in this described in the later are the following.

- We propose a new formalization for the subsequence anomaly detection problem, which overcomes the shortcomings of existing models. Our formalization is based on the intuitive idea that anomalous are the subsequences that are not similar to the common behavior, which we call normal.
- We describe a novel low-dimensional embedding for subsequences, and a corresponding graph representation. This representation leads to a natural distinction between recurring subsequences that constitute normal behavior, and rarely occurring subsequences that correspond to anomalies.
- Based on this representation, we develop Series2Graph, an unsupervised method for domain agnostic subsequence anomaly detection. Series2Graph supports the identification of previously unseen single and recurring anomalies, and can be used to find anomalies of different lengths.
- Finally, we conduct an extensive evaluation using several large and diverse datasets from various domains that demonstrates the effectiveness and efficiency of Series2Graph.

REFERENCES

[1] D. Abboud, M. Elbadaoui, W.A. Smith, and R.B. Randall. 2019. Advanced bearing diagnostics: A comparative study of two powerful approaches. *MSSP* 114 (2019).

[2] Jerome Antoni and Pietro Borghesani. 2019. A statistical methodology for the design of condition indicators. *Mechanical Systems and Signal Processing* (2019).

[3] Anthony J. Bagnall, Richard L. Cole, Themis Palpanas, and Konstantinos Zoumpatianos. 2019. Data Series Management (Dagstuhl Seminar 19282). *Dagstuhl Reports* 9, 7 (2019).

[4] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. R. Smith, Vicky Kalogera, and Aggelos K Katsaggelos. 2018. Machine learning for Gravity Spy: Glitch classification and dataset. *Information Sciences* 444 (1 5 2018), 172–186. https://doi.org/10.1016/j.ins.2018.02.068

[5] Paul Boniol and Themis Palpanas. 2020. GraphAn: Graph-based Subsequence Anomaly Detection. *PVLDB* (2020).

[6] Paul Boniol and Themis Palpanas. 2020. Series2Graph: Graph-based Subsequence Anomaly Detection for Time Series. *PVLDB* 13, 11 (2020).

[7] Nassia Daouayry, Ammar Mechouche, Pierre-Loic Maisonneuve, Vasile-Marian Scuturici, and Jean-Marc Petit. 2019. Data-Centric Helicopter Failure Anticipation: The MGB Oil Pressure Virtual Sensor Case. IEEE BigData.

[8] Medina Hadjem, Farid Naït-Abdesselam, and Ashfaq A. Khokhar. 2016. ST-segment and T-wave anomalies prediction in an ECG data using RUSBoost. In *Healthcom.*

[9] Katsiaryna Mirylenka, Alice Marascu, Themis Palpanas, Matthias Fehr, Stefan Jank, Gunter Welde, and Daniel Groeber. 2013. Envelope-Based Anomaly Detection for High-Speed Manufacturing Processes. *European Advanced Process Control and Manufacturing Conference* (2013).

[10] Themis Palpanas and Volker Beckmann. 2019. Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). *ACM SIGMOD Record* 48, 3 (2019).

[11] Tuomas Pelkonen, Scott Franklin, Paul Cavallaro, Qi Huang, Justin Meza, Justin Teller, and Kaushik Veeraraghavan. 2015. Gorilla: A Fast, Scalable, In-Memory Time Series Database. *PVLDB* 8, 12 (2015), 1816–1827. https://doi.org/10.14778/2824032.2824078

[12] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P. Boedihardjo, Crystal Chen, and Susan Frankenstein. 2015. Time series anomaly discovery with grammar-based compression. In *EDBT.*

[13] Li Wei, Eamonn J. Keogh, and Xiaopeng Xi. 2006. SAXually Explicit Images: Finding Unusual Shapes. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China.* 711–720. https://doi.org/10.1109/ICDM.2006.138

[14] Dragomir Yankov, Eamonn J. Keogh, and Umaa Rebbapragada. 2008. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. *Knowl. Inf. Syst.* 17, 2 (2008), 241–262.

[15] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn J. Keogh. 2016. Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *ICDM.* 1317–1322.

Lineage-Preserving Anonymization of the Provenance of Collection-Based Workflows

Khalid Belhajjame
PSL, Université Paris-Dauphine, LAMSADE
Paris, France
khalid.belhajjame@dauphine.fr

WORKFLOW AND PROVENANCE

Automated workflows have been shown to facilitate and accelerate scientific data exploration and analysis in many areas of sciences [9]. Figure 1 illustrates a simple workflow that is used to establish correlations between smoking and health conditions. Workflow provenance information, recorded during workflow executions, facilitates the interpretation of the results delivered by workflow execution. Beyond verification, workflow provenance information represents a useful dataset on its own right, that can be leveraged to answer queries that are relevant for an experiment that is (possibly related but) different from the original experiment, to learn new hypotheses, or to gain insight on the characteristics and quality of the data generated by given data modules. Collected workflow provenance information can also be used to respond to the requirements of funding agencies that are increasingly requesting the publication of the data generated in the context of research investigations.

In fields such as biomedicine and social sciences, workflow executions manipulate and generate sensitive information about individuals. To promote the publication and sharing of the provenance of workflow executions, we set out in this paper to examine the problem of anonymizing workflow provenance.

RELATED WORK

Related work has focused on the problem of securing workflow provenance and policing their access. For example, Chebotko *et al* [7] and Biton *et al* [6] proposed solutions that derive a partial view on a workflow provenance by hiding the data records of given modules Our objective is different from the above line of work in that we seek to provide the user with the provenance of all the modules of the workflow by leveraging anonymization.

Davidson *et al.* [10] investigated the problem of module privacy, whereby some of the parameters (attributes) characterizing the inputs and outputs of the modules are hidden to guarantee the privacy of modules. In our work, we seek, instead, to guarantee the privacy of the data records used and generated by the modules, instead of the behavior of the module.

We have examined in a previous workshop paper, the problem of identification of the k-anonymity degree that needs to be enforced when anonymizing the datasets used and generated by workflows [5]. In doing, we did not examine the problem of actually anonymizing workflow provenance. More importantly, we assumed that the modules that compose the workflow are 1-to-1 in that they produce

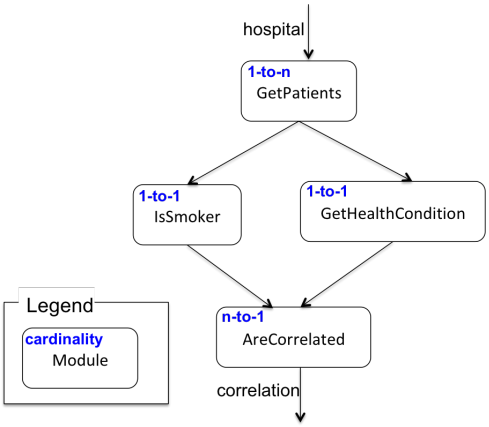


Fig. 1: Example workflow.

a single data record, given a single data record, and we did not give much thought to the problem of lineage preservation. In this paper, we are interested in what we refer to as collection-based workflows [12–14, 20]. The modules that compose such workflows can take as input a collection of data records and deliver a collection of data records. Such workflows have been advocated as a way to better meet the needs of non-expert users to model scientific data [15], and to structure complex relationships among related pieces of information that are processed together by the workflow [16]. This class of workflows has been overlooked in the literature w.r.t. privacy.

Different techniques have been proposed in the literature for protecting the privacy of individuals, notably, k-anonymity [19] and differential privacy [11]. In particular, differential privacy [11] has recently gained momentum as the method of choice in statistical databases. It involves adding random noise to the data so that the distribution of the resulting dataset is almost invariant to the inclusion of any data record. While powerful, differential privacy is not suitable for our purposes. It assumes that the user knows up-front the queries s/he wants to issue prior to the anonymization. This is not the case in our setting, where the scientist issues exploratory queries for understanding and eventually interpreting the results of the workflows. Furthermore, for it to be useful, the scientist should be able to inspect individual data records and their relationships (lineage), both of which are not possible using differential privacy. Indeed, differential privacy is more suited for statistical (i.e., aggregation-based) queries.

For our work, we chose to use k-anonymity [19]. This method is not as powerful as differential privacy when it comes to privacy guarantees. Yet, it is better suited for our purposes since it

can be instrumented, as we will show, to allow users to query and examine individual data records and their lineage within workflow provenance. k-anonymity is also still perceived by practitioners as sufficient for mitigating risk while maximizing utility, and real-world applications still utilize it for data sanitization (see e.g., [3, 8]). It is also widely popular and is used, e.g., in the healthcare world [1, 18], and is still recommended by data protection agencies (see e.g., [2]). This technique has been extensively investigated in the database and data mining communities [21]. Most of the proposals have focused on anonymizing a single relational table. In workflow provenance, however, we need to anonymize different datasets considering and preserving lineage relationships between them. One solution that can be used to anonymize workflow using k-anonymity would be to create a global relational table that is obtained by joining relations representing the input and output data records of the modules that compose the workflows. However, this solution suffers from the following issues. First, information about the same individual can be found in different records. This is because we consider collection based modules, e.g., a patient can be associated with multiple practitioners. Second, the same tuple in the global table may contain information about multiple individuals, e.g., a patient, one of its practitioners, etc. Moreover, as we will see later, different kinds of individuals may be associations with different k-anonymity degrees. For example, the k-anonymity degree associated with patients may be higher than that associated with practitioners. Traditional k-anonymity is not equipped to deal with the above issues. In this respect, the proposal by Nergiz *et al.* [17] is related to ours. They elaborated a technique that anonymizes multiple relations of a given database schema. While useful, this proposal makes a number of limiting assumptions. In particular, they consider snowflake schemas, in which there is a single relational table that represents individuals with the remaining relations containing quasi-attributes and having a single foreign key. In our work, we drop these assumptions and show that the anonymization of workflow provenance can be achieved in the presence of multiple datasets representing individuals with multiple relationships (foreign keys constraints) between them.

CONTRIBUTIONS

Our first contribution is the formulation of the problem of k-anonymization of the provenance of collection-based workflows. This is, to our knowledge, the first paper that extends the notion of k-anonymization from a single relation to the provenance of workflows. Our second contribution is a technique for k-anonymizing the provenance of a single module, i.e., input and output records together with their lineage information. Indeed, lineage information tracing the dependencies between the output and input of a module (and more generally a workflow) is key for third-party scientists to understand and examine the validity of workflow results. We examine this problem for modules that use and generate collections of data records. Our third contribution extends the technique proposed to cater for the anonymization of the provenance of a workflow as a whole. Central to the solution we present is the notion of k-group anonymity, which we define based on the k-anonymity degree and the magnitude of the smallest input (or output) set of data records used and generated by a module. This concept allows us to gracefully reason over the different k-anonymity degrees that may be associated with the inputs and outputs of the workflow's modules. We also show how the NP-hard problem of identifying the sets of data records to be grouped together into equivalence classes that meet k-anonymity requirements can be cast as a scheduling problem that we solve using integer programming.

The full contribution was published in the proceedings of the 23rd International Conference on Extending Database Technology [4]. As well as describing in details the above contributions, we address in the full paper an issue that is inherent to our anonymization technique, namely grouping sets of data records, and cast it as a scheduling problem. We also report on evaluation exercises that we empirically conducted to assess the effectiveness and efficiency of our solution.

REFERENCES

[1] K. Abouelmehdi, A. B. Hssane, and H. Khaloufi. Big healthcare data: preserving security and privacy. *J. Big Data*, 5:1, 2018.
[2] AEPD. k-anonymity as a privacy measure. *Spanish Agency for Data Protection*, 2018. https://www.aepd.es/media/notas-tecnicas/nota-tecnica-kanonimidad-en.pdf.
[3] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, and L. Murphy. A systematic comparison and evaluation of k-anonymization algorithms for practitioners. *Trans. Data Privacy*, 7(3):337–370, 2014.
[4] K. Belhajjame. Lineage-preserving anonymization of the provenance of collection-based workflows. In A. Bonifati, Y. Zhou, M. A. V. Salles, A. Böhm, D. Olteanu, G. H. L. Fletcher, A. Khan, and B. Yang, editors, *Proceedings of the 23rd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020*, pages 229–240. OpenProceedings.org, 2020.
[5] K. Belhajjame, N. Faci, Z. Maamar, V. A. Burégio, E. Soares, and M. Barhamgi. Privacy-preserving data analysis workflows for escience. In *EDBT/ICDT Workshops*, volume 2322 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.
[6] O. Biton, S. C. Boulakia, and S. B. Davidson. Zoom*userviews: Querying relevant provenance in workflow systems. In *VLDB*. ACM, 2007.
[7] A. Chebotko, S. Chang, et al. Scientific workflow provenance querying with security views. In *WAIM*, pages 349–356. IEEE CS, 2008.
[8] C. Clifton and T. Tassa. On syntactic anonymity and differential privacy. *Transactions on Data Privacy*, 6(2):161–183, 2013.
[9] R. F. da Silva et al. Automating environmental computing applications with scientific workflows. In *e-Science*, pages 400–406. IEEE, 2016.
[10] S. B. Davidson, S. Khanna, T. Milo, et al. Provenance views for module privacy. In *PODS*, pages 175–186, 2011.
[11] C. Dwork. Differential privacy. In *ICALP*, pages 1–12. Springer, 2006.
[12] R. Filgueira, A. Krause, M. P. Atkinson, et al. dispel4py: An agile framework for data-intensive escience. In *e-Science*, pages 454–464. IEEE, 2015.
[13] E. Griffis, P. Martin, and J. Cheney. Semantics and provenance for processing element composition in dispel workflows. In *WORKS*. ACM, 2013.
[14] J. Hidders, N. Kwasnikowska, J. Sroka, J. Tyszkiewicz, and J. V. den Bussche. DFL: A dataflow language based on petri nets and nested relational calculus. *Inf. Syst.*, 33(3):261–284, 2008.
[15] T. M. McPhillips, S. Bowers, D. Zinn, and B. Ludäscher. Scientific workflow design for mere mortals. *FGCS*, 25(5):541–551, 2009.
[16] P. Missier, N. W. Paton, and K. Belhajjame. Fine-grained and efficient lineage querying of collection-based workflow provenance. In *EDBT*. ACM, 2010.
[17] M. E. Nergiz, C. Clifton, and A. E. Nergiz. Multirelational k-anonymity. *IEEE Trans. Knowl. Data Eng.*, 21(8):1104–1117, 2009.
[18] N. Park, M. Mohammadi, K. Gorde, et al. Data synthesis based on generative adversarial networks. *PVLDB*, 11(10):1071–1083, 2018.
[19] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *PODS*, page 188. ACM Press, 1998.
[20] J. Sroka, J. Hidders, P. Missier, and C. A. Goble. A formal semantics for the taverna 2 workflow model. *J. Comput. Syst. Sci.*, 76(6):490–508, 2010.
[21] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. *VLDB Endowment*, 1(1):115–125, 2008.

Overlapping Hierarchical Clustering (OHC)

Ian Jeantet
Univ Rennes, CNRS, IRISA
Rennes, France
first.last@irisa.fr

Zoltan Miklos
Univ Rennes, CNRS, IRISA
Rennes, France
first.last@irisa.fr

David Gross-Amblard
Univ Rennes, CNRS, IRISA
Rennes, France
first.last@irisa.fr

1 INTRODUCTION

Agglomerative hierarchical clustering methods are widely used to analyze large amounts of data. These successful methods construct a dendrogram – a tree structure – that enables a natural exploration of data which is very suitable even for non-expert users. Various tools offer intuitive top-down or bottom-up exploration strategies, zoom-in and zoom-out operations, etc.

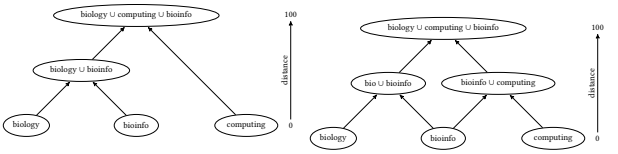
In the following real-life scenario where a social science researcher would like to understand the structure of specific scientific domains based on a large corpus of publications, such as dblp or Wiley, a contemporary approach is to to map keyterms in a into a high-dimensional space using word embedding [3] (for the sake of simplicity, we omit interesting issues such as preprocessing, polysemy, etc.). Identifying for example the denser regions in this space directly leads to insights on the key terms of Science. Moreover, building a dendrogram (Figure 1a) of key terms using an agglomerative method is typically used [1, 2] to organize terms into hierarchies.

Despite its usefulness, the dendrogram structure might be limiting. Indeed, any embedding of key terms has a limited precision, and key terms proximity is a debatable question. With classical agglomerative clustering, a merging decision has to be made, even if the advantage of one cluster on another is very small. Let us suppose that arbitrarily, *biology* and *bioinformatics* are merged. This may suggest to our analyst (not expert in computer science) that *bioinformatics* is part of *biology*, and its link to *computing* may be underestimated. Clearly, an interesting part of information is lost in this process.

In this paper, our goal is to combine the advantages of hierarchies while avoiding early cluster merge. This way, we deviate from the strict notion of trees, and produce a directed acyclic graph that we call a quasi-dendrogram (Figure 1b).

Our contributions are the following:

- We propose an agglomerative clustering method using a density-based merging condition that produces a directed acyclic graph of clusters instead of a tree, called a quasi-dendrogram,
- We introduce a new similarity measure to compare our method with other, quasi-dendrogram or tree-based ones,
- In the full paper we show through extensive experiments on real and synthetic data that we obtain high quality results with respect to classical hierarchical clustering, with reasonable time and space complexity.



(a) A classical dendrogram, hiding the early relationship between *bioinfo* and *computing*.

(b) A quasi-dendrogram, preserving the relationships of *bioinfo*.

Figure 1: Dendrogram and quasi-dendrogram for the structure of Science.

2 COMPUTING HIERARCHIES WITH OVERLAPS

Our algorithm, called OHC, computes a hierarchy of clusters that we can identify in the data. We call the generated structure a quasi-dendrogram and it is defined as follows.

Definition 2.1 (Quasi-dendrogram). A quasi-dendrogram is a hierarchical structure, represented as a DAG, where the nodes are clusters, such as:

- The leaves (level 0) contain a unique data point.
- There is only one root node that contains all the data points.
- Each non-root node has one or more parent nodes and a node is the union of its children.
- The nodes at a level δ represent a set of clusters that is a cover of all the data points.

The OHC method works as presented in Algorithm 1. We first compute the distance matrix of the data points (I3). We chose the cosine distance, widely use in NLP. Then we construct and maintain the δ-neighbourhood graph $G_\delta(V, E)$, starting from $\delta = 0$ (I4).

We also initialize the set of clusters, i.e. the leaves of our quasi-dendrogram, with the individual data points (I4). At each iteration, we increase δ (I6) and consider the new added links to the graph (I8) and the impacted clusters (I9). We extend these clusters by integrating the most linked neighbour vertices if the density does not change more than a given threshold λ (I10-15). We remove all the clusters included in these extended clusters (I16) and add the new set of clusters to the hierarchy as a new level (I18). We stop when all the points are in the same cluster which means that we reached the root of the quasi-dendrogram.

3 A HIERARCHY SIMILARITY MEASURE

As there is no ground truth on the hierarchy of the data we used, we need a similarity measure to compare the hierarchical structures produced by hierarchical clustering algorithms. The goal is not only to compare the topology but also the content of the nodes of the structure. First we construct a similarity between two given levels

Algorithm 1 Overlapping Hierarchical Clustering (OHC)

1: Input:
- $V = \{x_1, \ldots, x_N\}$, N data points.
- $\lambda \geq 0$, a merging density threshold.

2: Output: quasi-dendrogram H.
3: Preprocessing: obtain $\Delta = (\delta_1, \ldots, \delta_m)$ the distances between data points in increasing order.
4: Initialization:
- Create the graph $G(V, E_0 = \emptyset)$.
- Set a list of clusters $C = [\{x_1\}, \ldots, \{x_N\}]$.
- Add the list of clusters to the level 0 of H.

5: i=1.
6: **while** $\#C > 1$ and $i \leq m$ **do**
7: **for each** pair $(u, v) \in V^2$ such as $d(u, v) = \delta_i$ **do**
8: Add (u, v) to $E_{\delta_{i-1}}$.
9: Determine the impacted clusters C_{imp} of C containing either u or v.
10: **for each** impacted cluster $C_{imp_j} \in C_{imp}$ **do**
11: Look for the points $\{p_1, \ldots, p_k\}$ that are the most linked to C_{imp_j} in G_{δ_i}.
12: Compute the density $dens(S_j)$ of the subgraph $S_j = C_{imp_j} \cup \{p_1, \ldots, p_k\}$.
13: **if** $S_j \neq C_{imp_j}$ and $|dens(S_j) - dens(C_{imp_j})| \leq \lambda$ **then**
14: Continue to add the most linked neighbors to S_j the same way if possible.
15: When S_j stops growing remove C_{imp_j} from the list of clusters C and add S_j to the list of new clusters C_{new}.
16: Remove all cluster of C included in one of the clusters of C_{new}.
17: Concatenate C_{new} to C.
18: Add the list of clusters to the level δ_i of H.
19: i=i+1.
20: **return** H

of the hierarchies, and then we extend it to the global structures by exploring all the existing levels.

3.1 Level similarity

Given two levels l_1 (resp. l_2) of i clusters of the hierarchy h_1 and h_2 and the pairwise similarity matrix between their clusters, we can compute the similarity between l_1 and l_2 by taking the average of the maximal value for each row. Hence, the similarity function between two sets of clusters l_1, l_2 is defined as:

$$sim_l(l_1, l_2) = mean\{max\{J(c_1, c_2) \mid c_2 \in l_2\} | c_1 \in l_1\} \quad (1)$$

where J is the Jaccard similarity function.

However, taking the maximal value of each row shows how the clusters of the first set are represented in the second. If we take the maximal value of each column we will see the opposite, i.e. how the second set is represented in the first set. Hence with this definition the similarity might not be symmetrical so we propose this corrected similarity measure that shows how both sets are represented in the other one:

$$sim^*_l(l_1, l_2) = mean(sim_l(l_1, l_2), sim_l(l_2, l_1)) \quad (2)$$

3.2 Complete similarity

Now that we can compare two levels of the hierarchical structures, we can simply average the similarity for each corresponding levels of the same size. For classical dendrograms, each level has a distinct number of clusters so identification of levels is easy. Conversely, our quasi-dendrograms may have several distinct levels (pseudo-levels) with the same number of clusters. If so, we need to find the best similarity between these pseudo-levels. A matching M (Figure 2) should maximize the similarity between pseudo-levels while preserving their hierarchical relationship.

To produce this mapping, our simple algorithm is the following. We initialize M and two pointers with the two highest pseudo-levels

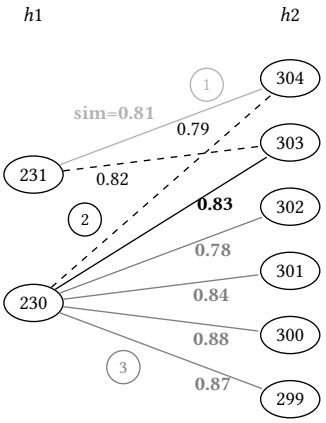


Figure 2: Computing the similarity between two quasi-dendograms h_1 and h_2 for levels having the same number of clusters.

$((l_1^{231}, l_2^{304})$, step 1 of Fig. 2). At each step, for each hierarchy, we consider current pointers and their children, and compute all their similarities (step 2). We then add pseudo-levels with maximal similarity to M (here, (l_1^{230}, l_2^{303})). Whenever a child is chosen, the respective pointer advances, and at each step, at least one pointer advances. Once pseudo-levels have been consumed on one side, ending with l, we can finish the process by adding (l^f, l) to M for all remaining pseudo-level l' on the other side (here, $l = l_1^{230}$. On our example, the final matching is $M = \{(l_1^{231}, l_2^{304}), (l_1^{230}, l_2^{303}), (l_1^{230}, l_2^{302}), (l_1^{230}, l_2^{301})$ $, (l_1^{230}, l_2^{300}), (l_1^{230}, l_2^{299})\}$.

Finally, We define the similarity between two hierarchies as

$$sim(h_1, h_2) = mean\{sim^*_l(l_1, l_2) | (l_1, l_2) \in (h_1, h_2) \ \& \ (l_1, l_2) \in M\}. \quad (3)$$

4 CONCLUSION AND FUTURE WORK

We propose an overlapping hierarchical clustering framework. We construct a quasi-dendrogram hierarchical structure to represent the clusters that is however not necessarily a tree (of specific shape) but a directed acyclic graph. In this way, at each level, we represent a set of possibly overlapping clusters. If the clusters present in the data show no overlaps, the obtained clusters are identical to the clusters we can compute using agglomerative clustering methods. In case of overlapping and nested clusters, however, our method results in a richer representation that can contain relevant information about the structure of the clusters of the underlying dataset.

REFERENCES

[1] David Chavalarias and Jean-Philippe Cointet. 2013. Phylomemetic patterns in science evolution - the rise and fall of scientific fields. *PloS one* 8, 2 (2013), e54847.
[2] Laércio Dias, Martin Gerlach, Joachim Scharloth, and Eduardo G Altmann. 2018. Using text analysis to quantify the similarity and evolution of scientific disciplines. *Royal Society open science* 5, 1 (2018), 171545.
[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

Ensuring License Compliance in Federated Query Processing

Benjamin Moreau
Nantes University, LS2N, CNRS
OpenDataSoft
Benjamin.Moreau@ls2n.fr
Benjamin.Moreau@opendatasoft.com

Patricia Serrano-Alvarado
Nantes University, LS2N, CNRS
Patricia.Serrano-Alvarado@ls2n.fr

KEYWORDS

Licenses, Federated Query, Privacy, Linked Data, RDF, Query Relaxation

1 INTRODUCTION AND MOTIVATION

A *federated SPARQL query* can retrieve information from several RDF data sources distributed across the Linked Data.

To facilitate reuse, data owners should systematically associate licenses with resources before sharing or publishing them[3, 14]. Licenses specify precisely the conditions of reuse of data, i.e., what actions are permitted, obliged, and prohibited.

When two or more licensed data sources participate in the evaluation of a federated query, the query result must be protected by a license such that each license of involved datasets is compatible with it. A license l_i is compatible with a license l_j if a resource licensed under l_j can be licensed under l_i without violating conditions of l_j.

Unfortunately, it is not always possible to find such a license[9]. In this case, the query result should not be reused nor published. We consider that a query whose result set cannot be licensed should not be executed.

Consider datasets of LargeRDFBench[12], a benchmark with 32 queries for federated query processing. Figure 1 shows the compatibility graph of Creative Commons licenses that protect LargeRDFBench datasets. The whole set of datasets of Figure 1 cannot be queried together because there is no license compliant with the fourth licenses. In this benchmark, 16 queries produce results that cannot be licensed.

One solution to the incompatibility of licenses is to negotiate with data providers to change a conflicting license, e.g., to ask DBpedia to change their license to CC BY or CC BY-NC. But negotiation takes time and is not always possible.

A second solution is to discard datasets that are protected by conflicting licenses. However, this solution can lead to a query with an empty result set. To face this problem, we use query relaxation techniques. That is to relax triple patterns to match triples of other datasets. But the number of possible relaxed queries can be huge and the least relaxed query may produce an empty result set. In a distributed environment, verifying each relaxed query is not feasible. So the challenge is to find the least relaxed query that returns a non-empty result while limiting communication costs.

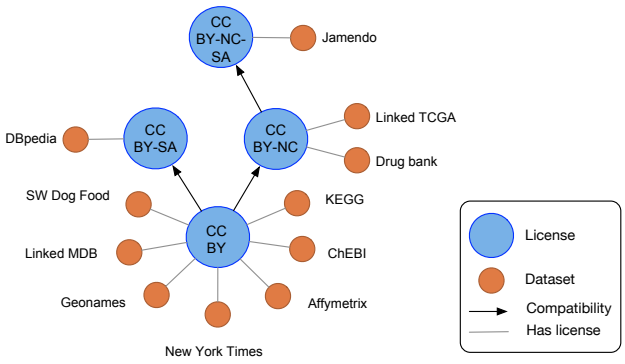


Figure 1: The compatibility graph of licenses for datasets of LargeRDFBench.

Our research question is, *given a SPARQL query and a federation of licensed datasets, how to guarantee a relevant and non-empty query result whose license is compliant with each license of involved datasets?*.

To our knowledge, there is no federated query engine that ensures license compliance. Many works focus on access control over linked data[1, 2, 6, 7, 10, 11]. These approaches do not resolve our problem because having the right to query datasets individually does not mean that it is possible to execute a federated query on them.

We propose FLiQue[1], a Federated License-aware Query processing strategy. FLiQue is designed to detect and prevent license conflicts and gives informed feedback with licenses able to protect a result set of a federated query. If necessary, it applies distributed query relaxation to propose a set of most similar relaxed queries whose result set can be licensed.

Our contributions are:

- a license-aware query processing strategy,
- an implementation of a license-aware federated query engine, and
- an experimental evaluation of our approach.

2 FLIQUE: A FEDERATED LICENSE-AWARE QUERY PROCESSING STRATEGY

We propose FLiQue, a federated license-aware query processing strategy to detect and prevent license conflicts in federated query

[1]In French, FLiQue is a homophone of *flic*, which means *cop*.

engines. FLiQue is located between the query parsing and the query optimization functions. It ensures that a query returning a non-empty licensed result set, i.e., a *candidate query*, is executed.

When the result set of a federated query cannot be licensed, we define sub-federations that avoid license conflicts. If there is no sub-federation able to produce a licensable and non-empty result set, we propose the least relaxed federated query for each sub-federation.

Compatibility graph of licenses. To know if a result set can be licensed, we need to know the license(s) with whom all licenses of datasets involved in a federated query are compatible. A compatibility graph of licenses contains a set of licenses partially ordered by compatibility. It can be defined by hand but licenses used in the Linked Data are not limited to well known licenses.

In this work, we use CaLi [8, 9], a lattice-based model for license orderings. It automatically orders a set of licenses in terms of compatibility. CaLi can provides all the licenses than should protect a result set and can also identify which licenses are in conflict.

If the result set of the original query cannot be licensed and any sub-federation can evaluate the original query, FLiQue finds the least relaxed query that can be evaluated on each sub-federation.

Query relaxation techniques. In this work, we use query relaxation using RDFS entailment rules as proposed in [5]. The idea consists of relaxation rules that use information from the ontology; these include relaxing a class to its super-class, a property to its super-property and, a term to a variable. The number of relaxed queries grows combinatorially with the number of relaxation rules, the richness of the ontology, and the relaxation possibilities of each triple pattern in the original query.

To avoid testing all relaxed queries, FLiQue executes them in a similarity-based ranking order, as in [4], until finding the candidate query. The *similarity measure* between a relaxed query and the original query is computed using statistical information about the concerned dataset, like the number of entities per class and the number of triples per property. However, the number of failing relaxed queries executed before finding the candidate query can be considerable.

FLiQue uses a strategy defined in [4]. Based on the source selection of the query engine, it identifies unnecessary relaxations generating failing relaxed queries. But, Computing similarity and identifying failing relaxed queries requires to communicate with data sources.

Data summaries. Some federated query engines, use statistics to reduce the communications to data sources during query processing, in particular in the source selection and query optimization steps.

FLiQue uses CostFed[13] source selection. It proposes data summaries, called *dataset capabilities*, containing datasets statistics such as the distinct properties with all the URI authorities of their subjects and objects prefixes. CostFed succeed in selecting relevant sources with precision while minimizing communication cost. Moreover, dataset capabilities can provide statistics to compute similarity measure.

3 CONCLUSION

In this work, we propose FLiQue, a federated license-aware query processing strategy. It ensures that a license protects the result set of any SPARQL query. To our knowledge, this is the first work that uses query relaxation in a distributed environment. Our implementation extends an existing federated query engine with our license-aware query processing strategy. The source code is available on GitHub under MIT license[2].. Our prototype demonstrates the usability of our approach. Experimental evaluation shows that FLiQue ensures license compliance with a limited overhead in terms of execution time. FLiQue is a step towards facilitating and encouraging the publication and reuse of licensed resources in the Web of Data. FLiQue is not a data access control strategy. It empowers well-intentioned data users in respecting the licenses of datasets involved in a federated query.

REFERENCES

[1] Luca Costabello, Serena Villata, and Fabien Gandon. 2012. Context-Aware Access Control for RDF Graph Stores. In *European Conference on Artificial Intelligence (ECAI)*.

[2] Alban Gabillon and Léo Letouzey. 2010. A View Based Access Control Model for SPARQL. In *International Conference on Network and System Security (NSS)*.

[3] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2020. Knowledge Graphs. *CoRR* abs/2003.02320 (2020).

[4] Hai Huang, Chengfei Liu, and Xiaofang Zhou. 2012. Approximating Query Answering on RDF Databases. *Journal of World Wide Web* 15 (2012).

[5] Carlos A Hurtado, Alexandra Poulovassilis, and Peter T Wood. 2008. Query Relaxation in RDF. *Journal on Data Semantics X* (2008).

[6] Yasar Khan, Muhammad Saleem, Aftab Iqbal, Muntazir Mehdi, Aidan Hogan, Axel-Cyrille Ngonga Ngomo, Stefan Decker, and Ratnesh Sahay. 2014. SAFE: Policy Aware SPARQL Query Federation Over RDF Data Cubes. In *Semantic Web Applications and Tools for Life Sciences (SWAT4LS)*.

[7] Sabrina Kirrane, Ahmed Abdelrahman, Alessandra Mileo, and Stefan Decker. 2013. Secure Manipulation of Linked Data. In *International Semantic Web Conference (ISWC)*.

[8] Benjamin Moreau, Patricia Serrano-Alvarado, Matthieu Perrin, and Emmanuel Desmontils. 2019. A License-Based Search Engine. In *Extended Semantic Web Conference (ESWC), Demo*.

[9] Benjamin Moreau, Patricia Serrano-Alvarado, Matthieu Perrin, and Emmanuel Desmontils. 2019. Modelling the Compatibility of Licenses. In *Extended Semantic Web Conference (ESWC)*.

[10] Said Oulmakhzoune, Nora Cuppens-Boulahia, Frédéric Cuppens, Stephane Morucci, Mahmoud Barhamgi, and Djamal Benslimane. 2014. Privacy Query Rewriting Algorithm Instrumented by a Privacy-Aware Access Control Model. *Annals of Telecommunications* 69 (2014).

[11] Pavan Reddivari, Tim Finin, Anupam Joshi, et al. 2007. Policy-Based Access Control for an RDF Store. In *Workshop Semantic Web for Collaborative Knowledge Acquisition (SWeCKa) collocated with IJCAI*.

[12] Muhammad Saleem, Ali Hasnain, and Axel-Cyrille Ngonga Ngomo. 2018. LargeRDFBench: a Billion Triples Benchmark for Sparql Endpoint Federation. *Journal of Semantic Web* 48 (2018).

[13] Muhammad Saleem, Alexander Potocki, Tommaso Soru, Olaf Hartig, and Axel-Cyrille Ngonga Ngomo. 2018. CostFed: Cost-Based Query Optimization for SPARQL Endpoint Federation. In *International Conference on Semantic Systems (SEMANTICS)*.

[14] Oshani Seneviratne, Lalana Kagal, and Tim Berners-Lee. 2009. Policy-Aware Content Reuse on the Web. In *International Semantic Web Conference (ISWC)*.

[2] github.com/benjimor/FLiQue

Confidentialité différentielle à risque : Relier les sources d'aléa et un budget de confidentialité

Ashish Dandekar
DI ENS, ENS, CNRS, Université PSL
& Inria & National University of Singapore
Paris, France & Singapour, Singapour
ashishd@comp.nus.edu.sg

Debabrota Basu
Chalmers University of Technology
& Inria
Göteborg, Suède & Lille, France
debabrota.basu@inria.fr

Pierre Senellart
DI ENS, ENS, CNRS, Université PSL
& Inria & Institut Universitaire de France
Paris, France
pierre@senellart.com

Stéphane Bressan
National University of Singapore

Singapour, Singapour
steph@nus.edu.sg

Dwork et al. [2] quantifient le niveau ε de confidentialité dans la *confidentialité ε-différentielle* comme une borne supérieure sur la perte de confidentialité, dans le pire des cas, obtenue par un mécanisme préservant la confidentialité. De manière générale, un mécanisme préservant la confidentialité perturbe les résultats en y ajoutant une certaine quantité de bruit aléatoire. La calibration du bruit dépend de la sensitivité de la requête et du niveau de confidentialité spécifié. Dans un scénario du monde réel, un coordinateur des données doit spécifier un niveau de confidentialité qui atteint un compromis entre les besoins des utilisateurs et les contraintes monétaires de l'entreprise. Par exemple, Garfinkel et al. [3] rapportent les difficultés rencontrées en déployant la confidentialité différentielle comme définition de confidentialité par le bureau du recensement des États-Unis. Ils insistent sur le manque de méthodes analytiques pour choisir le niveau de confidentialité. Ils fournissent également des études empiriques qui montrent la perte d'utilité obtenue en utilisant des mécanismes préservant la confidentialité.

Nous adressons le dilemme d'un coordinateur de données de deux manières. Premièrement, nous proposons une quantification probabiliste des niveaux de confidentialité. La quantification probabiliste des niveaux de confidentialité fournit au coordinateur des données une façon de prendre des risques quantifiés, en respectant un niveau d'utilité des données. Nous nous référons à cette quantification probabiliste par le terme de *confidentialité à risque* (Définition 1). Nous dérivons également un théorème de composition qui met en œuvre la confidentialité à risque. Deuxièmement, nous proposons un modèle de coût qui relie le niveau de confidentialité à un budget monétaire. Ce modèle de coût aide le coordinateur des données à choisir un niveau de confidentialité contraint par un budget estimé, et vice-versa. La convexité du modèle de coût proposé assure l'existence d'une confidentialité à risque unique qui minimise le budget. Nous montrons que la composition avec une confidentialité à risque optimale fournit des garanties de confidentialité plus fortes que le théorème classique de composition avancée [2]. Finalement, nous illustrons notre travail par un scénario réaliste qui démontre par l'exemple comment le coordinateur des données peut éviter de surestimer le budget en utilisant le modèle de coût proposé pour la confidentialité à risque. Pour plus de détails, se référer à la version longue de cet article [1].

La quantification probabiliste des niveaux de confidentialité dépend de deux sources d'aléa : l'*aléa explicite* induit par la distribution de bruit et l'*aléa implicite* de la distribution génératrice de données. Souvent, ces deux sources sont couplées l'une à l'autre. Nous imposons des formes analytiques des deux sources d'aléa ainsi qu'une représentation analytique de la requête pour dériver une garantie de confidentialité. Le calcul de la quantification probabiliste est, en général, une tâche difficile. Bien qu'il existe des définitions multiples de confidentialité probabiliste dans la littérature [4, 5], il manque une quantification analytique qui relie l'aléa et le niveau de confidentialité d'un mécanisme préservant la confidentialité.

Définition 1 (Confidentialité à risque). *Pour une distribution génératrice de données $\mathcal{G}$, un mécanisme préservant la confidentialité $\mathcal{M}$, équipé d'une requête f et de paramètres Θ, satisfait la confidentialité ε-différentielle avec une* confidentialité différentielle *$0 \leqslant \gamma \leqslant 1$ si, pour tous $Z \subseteq Image(\mathcal{M})$ et x, y échantillonés de $\mathcal{G}$ tels que $x \sim y$:*

$$\Pr\left[\left|\ln \frac{\Pr(\mathcal{M}(f,\Theta)(x) \in Z)}{\Pr(\mathcal{M}(f,\Theta)(y) \in Z)}\right| > \varepsilon\right] \leqslant \gamma, \tag{1}$$

où la probabilité externe est calculée par rapport à l'espace de probabilités $\text{Image}(\mathcal{M} \circ \mathcal{G})$ *obtenu en appliquant le mécanisme préservant la confidentialité $\mathcal{M}$ sur la distribution génératrice de données $\mathcal{G}$.*

Autant que nous en le sachions, nous sommes les premiers à dériver une confidentialité à risque pour le mécanisme de Laplace [2], largement utilisé. Nous dérivons également un théorème de composition pour la confidentialité à risque. C'est un cas particulier du théorème de composition avancée [2] qui traite d'un usage séquentiel et adaptatif de mécanismes préservant la confidentialité. Ces résultats et leurs preuves sont disponibles dans [1].

Le niveau de confidentialité proposé par le cadre de la confidentialité différentielle est une quantité trop abstraite pour être intégrée dans un contexte d'affaires. Nous analysons et listons les conditions d'un modèle de coût qui transforme le niveau de confidentialité en un budget monétaire. Nous l'illustrons (équation (2))

en choisissant une fonction qui satisfait ces conditions. Dans l'équation (2), E dénote le budget de compensation qu'une entreprise doit payer à chaque partie prenante dans le cas d'une violation des données à caractère personnel quand les données sont traitées sans garantie de confidentialité prouvée, et E_ε^{cd} est le budget de compensation qu'une entreprise doit payer à chaque partie prenante dans le cas d'une violation des données à caractère personnel quand les données sont traitées par un mécanisme avec confidentialité ε-différentielle. $E_{\min}$ et c sont des hyper-paramètres réglables. Le lecteur pourra se référer à [1] pour plus d'explications.

$$E_\varepsilon^{cd} \triangleq E_{\min} + Ee^{-\frac{c}{\varepsilon}}. \quad (2)$$

La fonction choisie pour le modèle de coût pour la quantification probabiliste du modèle de coût est convexe en le niveau de confidentialité. Ainsi, elle conduit à un niveau de confidentialité probabiliste unique qui minimise le coût. Nous illustrons ceci par un scénario réaliste d'une entreprise respectant le RGPD qui a besoin d'une estimation du budget de compensation qu'elle devra payer aux parties prenantes dans le cas malheureux d'une violation de données personnelles. Cette illustration montre que l'usage des niveaux de confidentialité probabilistes évite de surestimer le budget de compensation sans sacrifier l'utilité.

Nous évaluons de plus les garanties de confidentialité en utilisant un calcul de la confidentialité à risque pour le mécanisme de Laplace. Nous comparons quantitativement la composition sous confidentialité à risque optimale, estimée avec les modèle de coût, avec les mécanismes traditionnels de composition – de base et avancé [2]. Nous observons de plus fortes garanties de confidentialité que celles obtenues par la composition avancée, sans sacrifier l'utilité du mécanisme. Nous adaptons également le système PATE [6], qui utilise la technique de comptabilité des moments de l'état de l'art, pour utiliser la confidentialité à risque. Nous montrons expérimentalement que la confidentialité à risque optimale fournit de meilleures garanties que la comptabilité des moments.

En conclusion, les bénéfices de la quantification probabiliste, c.-à-d., de la confidentialité à risque, sont doubles. Non seulement elle quantifie le niveau de confidentialité pour un mécanisme préservant la confidentialité donné, mais elle facilite également la prise de décision dans des problèmes qui se focalisent sur le compromis confidentialité–utilité et sur la minimisation du budget de compensation.

REMERCIEMENTS

Ce travail a été soutenu par la National Research Foundation (NRF) de Singapour, Corporate Laboratory@University Scheme, National University of Singapore et Singapore Telecommunications Ltd. Ces recherches ont également été financées par le projet BioQOP de l'ANR française (ANR-17-CE39-0006).

RÉFÉRENCES

[1] Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. 2021. Differential Privacy at Risk : Bridging Randomness and Privacy Budget. *Proceedings on Privacy Enhancing Technologies* 1 (2021). https://doi.org/10.2478/popets-2021-0005

[2] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[3] Simson L Garfinkel, John M Abowd, and Sarah Powazek. 2018. Issues Encountered Deploying Differential Privacy. *arXiv preprint arXiv :1809.02201* (2018).

[4] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. 2012. Random Differential Privacy. *Journal of Privacy and Confidentiality* 4, 2 (2012), 43–59.

[5] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. 2008. Privacy : Theory meets practice on the map. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on.* IEEE, 277–286.

[6] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.*

Guided Exploration of User Groups

Published in the proceedings of VLDB conference (April 2020)

Mariia Seleznova*
TU Berlin
seleznova@tu-berlin.de

Behrooz Omidvar-Tehrani
NAVER LABS Europe
behrooz.omidvar-tehrani@naverlabs.com

Sihem Amer-Yahia†
CNRS, Université of Grenoble Alpes
sihem.amer-yahia@univ-grenoble-alpes.fr

Eric Simon
SAP Paris
eric.simon@sap.com

1 INTRODUCTION

User data is widely available in various domains and is characterized by a combination of demographics such as age and location, and actions such as rating a movie, providing advice on a product, or recording one's blood pressure. Many companies address the very fast growing market of user data analysis by proposing dedicated platforms to collect and analyze such data in a variety of business segments, and start to tightly couple user data management with enterprise operational data management solutions [1, 2].

A common way of understanding user data is *user group analysis* whose purpose is to breakdown users into groups to gain a more focused understanding of their behavior or to identify a target group of users satisfying an information need. User group analysis has many applications in domains such as social sciences, product design, product marketing campaigns, and customer services. For instance, a data scientist may conduct large-scale population studies to gain insights on the preferences of various population segments. An information consumer may want to explore alike user groups to be inspired for routine tasks such as choosing a product or picking a service subscription. In [3], we apply user group analysis to the task of program committee formation, where a PC chair starts with any seed group of researchers, and looks iteratively for users in groups that match properties expected of a PC (geographic distribution, gender and topic balance, etc).

Due to the iterative nature of the task at hand, user group analysis can be viewed as an instance of Exploratory Data Analysis (EDA). In this setting, the exploratory analysis of user groups is an iterative decision-making process, whereby an analyst is shown a set of user groups labeled with user and item attribute values (e.g., groups of researchers who published in VLDB), chooses target users from those groups, and selects the best exploration action to move to the next iteration (e.g., select a group whose label is [prolific, female, published in VLDB] that has researchers with well distributed geographical location, and apply an exploration action to return k diverse groups that overlap with the selected group). Despite a large body of work on the recommendation of data exploration actions, several shortcomings exist. First, most works recommend SQL/OLAP queries, which is not adapted to analysts with no IT expertise, while other works focus on specific types of actions that are too limited for user group analysis. Second, existing solutions either rely on tedious manual exploration [4], or on a log of exploration sessions on the same dataset [5]. This assumption is not valid in our case since many analysis tasks are performed on different datasets. Finally, recent work on automating the exploration process does not generate interpretable exploration policies [6, 7].

We propose the first framework for user group exploration that learns an exploration policy *without requiring prior collection of exploration logs*. The learned policy is used to recommend an end-to-end interpretable exploration session on a given dataset. Our framework supports exploration sessions consisting of a sequence of exploration actions of various types and returns recommendations for the exploration of new datasets, provided they are similar (with respect to domain-specific features) to previously analyzed datasets. Our framework is *independent* from the approach used to generate user groups from raw data. The exploration actions operate on groups that can be generated with any method ranging from SQL aggregate queries to k-means, graph-based algorithms (e.g., community detection), and graph representation learning, to name a few. In an offline phase, we learn an exploration policy from a simulated agent experience in such a way that no human intervention is needed. The lack of real exploration logs from a variety of users rules out the use of approaches that require large amounts of exploration traces such as sequence mining. We summarize our main contributions below (C1 to C3). Further details are provided in [8].

C1: User group analysis as EDA. Our first contribution is to formalize the guided EDA problem applied to user groups as a Markov decision process where a state displays several groups, a transition is the application of an exploration action to a chosen group, and the reward of an action is a function of the number of target users discovered by that action. An exploration session is hence a sequence of exploration actions, and an exploration policy is a function that maps a state to an action. A common difficulty is the design of state features to capture the application of an exploration action to a group. To reflect a human agent, we propose semantics that unifies previously introduced actions [4]. State features are carefully designed to capture the effect of an action on a state. The problem of guided EDA on user groups becomes that of finding

*Work completed when author was an intern at SAP.

†Work funded by the Horizon 2020 research and innovation programme under grant agreement No 863410.

a policy that maximizes utility, i.e., that discovers as many target users as possible regardless of the dataset and the seed group.

C2: Learning the exploration policy. Our second contribution is the use of Reinforcement Learning (RL) to learn an exploration policy from a simulated agent experience, i.e., an RL agent that acts as an analyst who chooses exploration actions, and recommends a policy. A notable advantage of RL methods is that, unlike supervised methods, they do not require to gather labeled data. Instead, the agent learns from rewards computed by the environment during the interaction. This naturally fits our context since we can use, for instance, the PC of WebDB 2017 (or any previous PC of the same or different venue) to learn an exploration policy offline, and apply it online to build the PC of WebDB 2018. Additionally, the use of RL for EDA enables to model our problem as an interactive stochastic process, a nascent research area in databases, that can greatly benefit from research experience in data modeling. Our work differs from recent proposals on RL-based EDA [9]. First, we address a well-defined user data analysis task (gathering target users) and the exploration actions are chosen accordingly. Additionally, in our case, running a large number of manual explorations for each new dataset and new analysis task is impractical. Moreover, our focus on producing interpretable exploration sessions for a human analyst warrants the use of classical RL methods instead of deep RL or contextual bandits, a special case of RL where the agent's action does not determine the next state of the environment.

C3: Experiments. Our third contribution is an extensive set of experiments that validate the use of RL to solve the guided EDA problem for user groups. While we do not focus on theoretical convergence results, we empirically study various cases: users to be found are scattered in groups, exploration with different starting groups, and how our learned policies perform with respect to random ones. We also examine the utility of learned policies where they are learned on the same dataset or transferred between datasets. We also validate the appropriateness of the state features we crafted. Our experiments show that the simulated agent succeeds to learn an interpretable policy that naturally captures human analysts (as in [4]) and can hence be used to automate guided EDA.

2 SOLUTION

The general architecture for our RL-based approach is depicted in Figure 1. We apply the learning procedure in an offline phase and recommend the learned exploration policy online. The offline phase addresses our first challenge: simulate a human experience in such a way that we learn a policy that is applicable to any dataset and any initial group. During the offline phase, an agent simulating a human analyst is trained to learn a policy that maximizes utility, e.g., for the PC of WebDB 2017. The policy is updated as the agent interacts with user groups via exploration actions. The outcome is final exploration policy that can be leveraged in an online phase to find any set of target users. We model EDA as a Markov Decision Process (MDP) with states, exploration actions, and rewards for transitioning between the states. Our goal is to find the optimal policy which maximizes the cumulative reward. An optimal policy always selects actions with the highest value in the current state, thus maximizing expected reward. We solve our policy optimization

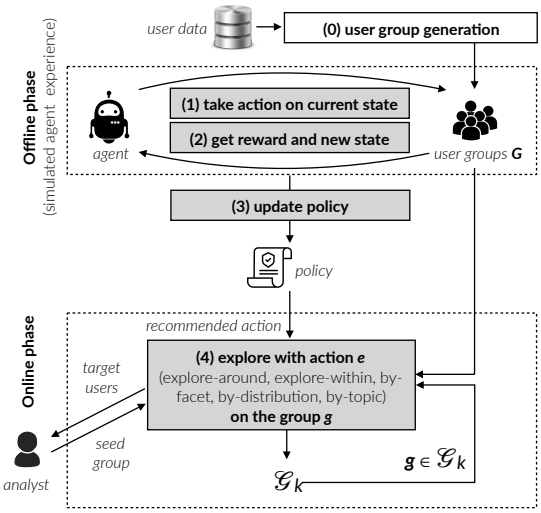


Figure 1: RL framework architecture. In the offline phase, the system iterates between steps 1 and 2 until it learns a policy. The policy (step 3) is used online to recommend exploration actions. In step 4, it is applied to any input seed group and returns target users.

problem with semi-gradient SARSA algorithm (described in [8]) based on a stochastic gradient descent (SGD) minimization of mean squared error.

To enable learning interpretable EDA policies, we describe our MDP states with a small set of domain-dependent features that reflect the result of applying an exploration action to a state. The features must enable learning how to choose between actions at each step. We then rely on approximate control methods [10] to blend together states with the same features and learn an approximation of the optimized policy.

REFERENCES

[1] Qualtrics Marketplace (SAP). https://www.qualtrics.com/marketplace/.
[2] Amplitude Behavioral Analytics Platform. https://amplitude.com/behavioral-analytics-platform/.
[3] Behrooz Omidvar-Tehrani and Sihem Amer-Yahia. Tutorial on user group analytics: Discovery, exploration and visualization. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2307–2308, 2018.
[4] Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, and Alexandre Termier. Interactive user group analysis. In *International Conference on Information and Knowledge Management (CIKM)*, pages 403–412, 2015.
[5] Tova Milo and Amit Somech. Next-step suggestions for modern interactive data analysis platforms. In *International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 576–585, 2018.
[6] Ori Bar El, Tova Milo, and Amit Somech. ATENA: an autonomous system for data exploration based on deep reinforcement learning. In *International Conference on Information and Knowledge Management (CIKM)*, pages 2873–2876, 2019.
[7] Ori Bar El, Tova Milo, and Amit Somech. Automatically generating data exploration sessions using deep reinforcement learning. In *International Conference on Management of Data (SIGMOD)*, 2020.
[8] Mariia Seleznova, Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, and Eric Simon. Guided exploration of user groups. *Proc. VLDB Endow.*, 13(9):1469–1482, 2020.
[9] Amit Somech, Tova Milo, and Chai Ozeri. Predicting "what is interesting" by mining interactive-data-analysis session logs. In *International Conference on Extending Database Technology (EDBT)*, pages 456–467, 2019.
[10] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Une Évaluation Comparative des Algorithmes de Recommandation : *cas d'étude sur les clients* TOTAL

Idir Benouaret
idir.benouaret@univ-grenoble-alpes.fr
CNRS, Univ Grenoble Alpes
Grenoble, France

Sihem Amer-Yahia
sihem.amer-yahia@univ-grenoble-alpes.fr
CNRS, Univ Grenoble Alpes
Grenoble, France

ABSTRACT

L'application des systèmes de recommandation a souvent pour objectif de recommander les top-N produits qui sont les plus pertinents pour les clients, et se focalise essentiellement sur les produits qui sont le plus susceptibles d'être achetés dans le futur proche. Dans cet article expérimental, nous présentons une évaluation extensive et expérimentale basée sur des données réelles d'achats fourni par notre partenaire industriel : TOTAL. Cette étude à pour but de comparer différentes approches d'algorithmes de filtrage collaboratif. Ces expérimentations sont partie intégrante du développement d'une campagne d'offres promotionnelles chez TOTAL. Nous montrons dans cet article comment différentes façons d'exploiter les données d'apprentissage et d'application les algorithmes de recommandation influencent les performances. Cet article dans sa version intégrale est publié dans les actes de IEEE BigData 2020 [1].

KEYWORDS

Systèmes de recommandation, évaluation, application

1 INTRODUCTION

Les systèmes de recommandation se sont avérés être un outil efficace pour les plateformes de e-commerce et de la grande distribution pour fidéliser leurs clients et augmenter leur profit. Dans cet article, nous présentons un cas d'étude expérimental sur une multitude d'algorithmes de filtrage collaboratif en utilisant les données qui nous sont fournies par notre partenaire industriel TOTAL. Le jeu de données contient plus de 440.000 clients possédant une carte de fidélité, sur une période de presque 3 ans et qui contient environ 3 millions d'achats (cf. Table 1).

2 ALGORITHMES DE RECOMMANDATION

Nous implémentons et évaluons quatre catégories d'algorithmes de filtrage collaboratif : les règles d'associations [3] (ARM), filtrage collaboratif basé sur les items [5] (IBCF), la factorisation matricielle [2] (Implicit-ALS) et le classement bayésien personnalisé [4] (BPRMF). Nous avons aussi implémenté une méthode de recommandation (Most-Pop) qui est utilisé chez TOTAL et qui consiste à recommander à chaque client les produits les plus populaires. La table 3 illustre les recommandations des 5 meilleurs produits pour le même client en utilisant les différentes méthodes de recommandations.

Table 1: Caractéristiques de notre jeu de données

	jeu de données TOTAL
Durée	Jan 2017 −>Sept 2019
Nombre de clients	442, 520
Nombre de produits	9, 366
Nombre d'achats	2, 833, 938

Table 2: Résultats en utilisant l'historique complet des achats.

Algorithme	$F1@10$	$DCG@10$
ARM	6.76%	55.77%
IBCF	**8.06%**	**63.71%**
Implicit-ALS	7.81%	60.05%
BPRMF	5.96%	42.91%
MostPop	6.02%	39.32%

3 EXPÉRIMENTATION

Dans cette section, nous présentons la comparaison de la performance des algorithmes de recommandation dans différents contextes.

3.1 Évaluation standard

Lorsque nous utilisons l'ensemble des données disponibles, nous constatons que IBCF est supérieur à toutes les autres approches de recommandation (cf. Table 2). Il est pertinent de souligner que nos résultats ne sont pas entièrement similaires avec ceux rapportés dans d'autres études, telles que [3] qui ont constaté que le modèle de règles d'association fait mieux que d'autres approches de filtrage collaboratif plus élaborées, y compris les modèles de factorisation matricielle. Cela implique qu'aucune conclusion générale ne peut être tirée sur la performance relative de chaque algorithme de recommandation sans mener des expériences approfondies.

3.2 Contribution de la Récence des Données

Pour examiner l'effet de la récence des achats sur précision, nous menons des expériences avec des données d'apprentissage de différentes durées: 18 derniers mois, 12 derniers mois et 6 derniers mois. Nos résultats montrent clairement que l'apprentissage sur toutes les données disponibles est inférieure à l'apprentissage uniquement sur les achats récents, ce qui nous montre l'importance d'intégrer la récence des données dans la génération de recommandations. Nos résultats suggèrent que l'historique d'apprentissage le plus récent (6 mois) donne des meilleures résultats (cf. Table 4).

3.3 Contribution des Facteurs Contextuels

Nous continuons notre exploration de l'effet du temps sur la précision des recommandations et menons des expériences qui divisent

Table 3: Historique d'achat d'un client et top-5 produits recommandés par chaque algorithme de recommandation sélectionné.

	Recommandations Top-5 de chaque algorithme pour le même client			
Historique d'achat du client	ARM	IBCF	Implicit-ALS	BPRMF
TOTAL deicer *Orangina 33cl* *Winter windscreen washer* *Ham & cheese Pizza* *Manhattan salad* *Engine oil 4tz* *Brake fluid hbf4* *Coca Cola 1.5L*	*Evian sparkling 1L* *TOTAL windshield washer* *Cristaline water* *TOTAL car wash* *Expresso*	*TOTAL windshield washer* *TOTAL car wash* *Cristaline sparkling* *Coca Cola 50cl* *Lg bug remover*	*TOTAL windshield washer* *Coca Cola 50cl* *Salad Roma 320* *Salad Antibes* *Evian water 75 CL*	*TOTAL car wash* *TOTAL Adblue* *TOTAL windshield washer* *Plastic wipes* *Lg bug remover*

Table 4: Résultats en variant la taille des données d'apprentissage

	Apprentissage avec 6 mois		Apprentissage avec 12 mois		Apprentissage avec 18 mois	
Algorithme	*F*1@10	DCG@10	*F*1@10	DCG@10	*F*1@10	DCG@10
ARM	9.04% (+33.77%)	60.54% (+8.55%)	8.01% (+18.49%)	57.89% (+3.80%)	7.64% (+13.01%)	57.06% (+2.31%)
IBCF	**10.47%**(+29.9%)	**72.94%**(+14.48%)	8.97% (+11.29%)	68.51% (+7.53%)	8.75% (+8.56%)	64.39% (+1.06%)
Implicit-ALS	8.35% (+6.91%)	65.95%(+9.82%)	7.92% (+1.4%)	63.17% (+5.19%)	8.6%(+10.11%)	62.87% (+4.68%)
BPRMF	7.42% (+23.48%)	45.11% (+5.12%)	6.14% (+3.02%)	44.86% (+4.54%)	6.47% (+8.55%)	44.16% (+2.91%)
MostPop	7.62% (+26.57)	43.62% (+10.93%)	6.74% (+11.96)	41.61% (+5.82%)	6.46% (+11.96)	40.43% (+2.82%)

Table 5: Résultats sur le contexte temporel

	Contexte = Matin				Contexte = Après-midi				Contexte = Soir			
	Non contextuel		Contextuel		Non contextuel		Contextuel		Non contextuel		Contextuel	
Algorithme	F1@10	DCG@10	F1@10	DCG@10	F1@10	DCG@10	F1@10	DCG@10	F1@10	DCG@10	F1@10	DCG@10
ARM	6.19%	43.22%	6.74%	47.73%	4.88%	40.86%	6.47%	45.59%	4.63%	34.74%	6.24%	38.85%
IBCF	7.03%	46,97%	**8.34%**	**50.48%**	5.79%	52.76%	**8.65%**	**59.05%**	6.73%	55.07%	**7.46%**	**58.98%**
Implicit-ALS	6.25%	44.53%	7.77%	49.97%	5.38%	47.66%	7.32%	52.68%	6.68%	52.77%	7.19%	54.63%
BPRMF	5.05%	32.27%	5.71%	41.04%	4.41%	29.62%	5.23%	39.15%	4.72%	23.74%	4.56%	31.28%
MostPop	4.76%	24.59%	5.10%	32.58%	3.82%	22.75%	4.92%	31.03%	2.84%	18.61%	4.17%	24.39%

Table 6: Résultats sur différents segments d'utilisateurs

Segment	nb d'achats	F1@10				
		ARM	IBCF	Implicit-ALS	BPRMF	MostPop
Rare	1	**7.03%**	6.19%	4.82%	5.23%	6.86%
	2	**7.96%**	7.15%	5.93%	5.36%	6.74%
	3	**8.07%**	7.16%	6.21%	6.11%	6.61%
	4	**8.6%**	8.08%	6.61%	6.76%	6.85%
Occasionnel	5-10	8.97%	**9.22%**	6.73%	5.9%	6.59%
	11-20	7.92%	**9.05%**	8.23%	6.12%	6.28%
Fréquent	21-40	7.05%	7.97%	**8.72%**	7.13%	5.62%
	>40	5.69%	6.67%	**7.78%**	5.61%	4.77%

l'ensemble d'apprentissage en matin, après-midi et soir. Nous observons que des améliorations significatives de la précision peuvent être obtenues en incorporant des informations contextuelles temporelles dans les algorithmes de recommandation (cf. Table 5). Des améliorations similaires peuvent être obtenues avec le contexte de localisation, où on construit un modèle local pour les achats projetés dans une région donnée (ex. Île-de-France) en comparaison à un modèle global qui est appris sur toutes les données.

3.4 Évaluation sur la Fréquence d'Achat

Nous évaluons aussi les performances des algorithmes sur différents segments de clients en fonction de leur fréquence d'achat. Nos résultats montrent une nette différence de performances suivant les segments construits (cf. Table 6). En particulier, nous montrons que pour les clients rares, le fait de s'appuyer sur l'extraction de règles d'association (ARM) offre la plus grande précision. Alors que, les clients fréquents sont mieux servis par les algorithmes de factorisation matricielle (Implicit-ALS). Pour les clients occasionnels, IBCF est le meilleur algorithme. Cela indique qu'aucune conclusion générale ne peut être tirée sur les performances relatives de chaque algorithme et que tester toutes les méthodes avec différents segments de clientèle est nécessaire.

REFERENCES

[1] Idir Benouaret and Sihem Amer-Yahia. 2020. A Comparative Evaluation of Top-N Recommendation Algorithms: Case Study with TOTAL Customers. In *Proceedings of the twenty-first international conference on Big Data.*

[2] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.

[3] Bruno Pradel, Savaneary Sean, Julien Delporte, Sébastien Guérif, Céline Rouveirol, Nicolas Usunier, Françoise Fogelman-Soulié, and Frédéric Dufau-Joel. 2011. A case study in a recommender system based on purchase data. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 377–385.

[4] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[5] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms. *Www* 1 (2001), 285–295.

Optimisation Collective d'Arbres de Décision dans une Forêt Alétoire

Nour ElIslem Karabadji
École Supérieure de Technologies Industrielles d'Annaba
Algerie

Hassina Séridi
LabGed, Badji Mokhtar-Annaba University
Algerie

Abdelaziz Amara Korba
LRS, Badji Mokhtar-Annaba University
Algérie

Sabeur Aridhi
LORIA, CNRS, Université de Lorraine
France

Wajdi Dhifli
Univ. Lille, CHU Lille, ULR 2694 - METRICS
France

RÉSUMÉ

La méthode d'ensemble des forêts aléatoires vise à concevoir un groupe d'arbres de décision construits sur la base d'un échantillonnage aléatoire sur les instances et les attributs d'apprentissage. Cette stratégie offre aux forêts aléatoires une forte capacité de généralisation. Cependant, il est primordial que pendant la construction du modèle, les arbres de décision construits soient précis (par rapport au taux de bon classement) et diversifiés (au niveau de leurs structures). Dans cet article, nous proposons une approche de construction d'une forêt aléatoire qui repose sur une optimisation collective de l'ensemble des arbres de décision du modèle. Le modèle proposé vise à : 1) trouver un bon taux de classification des arbres de décision du forêt aléatoire afin de maximiser la performance de classification du modèle ensembliste et 2) utiliser une mesure de diversité entre les arbres de décision afin d'améliorer la capacité de généralisation de la forêt aléatoire. Les analyses expérimentales effectuées sur plusieurs jeux de données montrent la supériorité de notre modèle en comparaison avec l'approche classique des forêts aléatoires ainsi que d'autres approches concurrentes.

1 INTRODUCTION

Bien que les arbres de décision soient toujours parmi les techniques de classification les plus populaires en raison de leur efficacité, simplicité et interprétabilité, ils sont considérés comme des classificateurs instables, car ils dépendent fortement du jeu de données d'apprentissage et sont sensibles à de nombreuses incertitudes dans les données (particularité, bruit, variation résiduelle, *etc*.)[4]. Cependant, il existe une méthode populaire qui permet de tirer pleinement profit des avantages des arbres de décision en les regroupant pour créer une forêt aléatoire. En pratique, un bon nombre d'arbres de décision sont générés sur la base d'un échantillonnage et d'une sélection aléatoire d'attributs [3]. Durant la construction d'une forêt aléatoire, il est important que les arbres de décision qui la composent soient à la fois précis (en terme de performance de classification) et diversifiés (au niveau de leurs structures). Ce processus de randomisation peut aider à la confrontation de ces deux défis. Toutefois, la méthode la plus utilisée dans la littérature pour ces types de problèmes consiste à construire et combiner un très grand nombre d'arbres de décision [1]. Une telle génération d'un grand nombre d'arbres entraîne une augmentation considérable des coûts de calcul et de consommation mémoire [2].

Dans cet article, nous proposons une approche d'optimisation à base d'un algorithme génétique pour la construction de forêt aléatoire, dont l'objectif est de garantir simultanément un taux élevé de bonne classification, la diversité des arbres générés et le nombre d'arbres dans la forêt aléatoire construite. Nous avons mené une étude empirique sur l'efficacité de l'approche proposée à travers une évaluation expérimentale sur 10 ensembles de données du référentiel d'apprentissage automatique UCI [5]. Les résultats obtenus montrent l'efficacité de notre approche en comparaison avec la méthode de base ainsi que d'autres approches concurrentes.

2 OPTIMISATION DE LA CONSTRUCTION D'UNE FORÊT ALÉATOIRE

La construction d'une forêt aléatoire consiste à générer k arbres de décision à la base d'un échantillonnage aléatoire avec remplacement de l'ensemble d'apprentissage TR et une sélection d'attributs. Le schéma d'échantillonnage avec remplacement est une sélection d'un sous-ensemble d'instances S_i composé de $n^{'}$ instances avec $n^{'} \leq |TR|$. Chaque instance de S_i peut apparaître plusieurs fois. La sélection d'attributs consiste à sélectionner aléatoirement un sous-ensemble A_i de l'ensemble total d'attributs A avec $A_i \subseteq A$. Les solutions possibles de combinaison de sous-ensemble S_i et A_i forment un treillis booléen$\mathcal{P}_{TR}/\mathcal{P}_A$ composé de tous les sous-ensembles de TR/A avec certaines relations d'ordre (c.-à-d., au moins une relation binaire $\subseteq$). De plus, pour la séquence de répétitions R_i représentant le nombre de répétitions de chaque instance dans S_i, les solutions possibles forment aussi un treillis booléen (compositions d'un entier positif) de l'ordre de $2^{|TR_i|}$. L'ensemble des treillis booléens sont organisés par niveau en fonction de la taille des sous-ensembles/séquence. De plus, les sous-ensembles/séquence de chaque niveau sont tous ordonnés (c.-à-d., ils forment une chaîne ordonnée). En accord avec ces deux ordres, chaque sous-ensemble $Y \subseteq X$ peut être facilement identifié à l'aide d'une paire d'indices (l_i, x_i) où l_i est la taille de Y et x_i est sa position dans le niveau. En d'autres termes, le couple (l_i, x_i) indique respectivement le niveau $|Y|$ et la position de Y au sein du niveau donné.

L'objectif principal de ce travail est de construire une forêt aléatoire optimisée en termes de taux de classification, du nombre

d'arbres dans la forêt et de leurs diversité. Cet objectif sera atteint en utilisant un groupe de couples appropriés d'échantillons et d'attributs d'apprentissage. Cet ensemble de couples doit être soigneusement sélectionné parmi de nombreuses possibilités. Chaque solution (une forêt aléatoire) peut être présentée comme un groupe G de k couples de jeux d'instances d'apprentissage et d'attributs, c.-à-d. $G = \{(TR_a, A_b)_1, (TR_b, A_c)_2, ..., (TR_i, A_j)_k\}$ où TR_i (c.-à-d. TR_a, TR_b, $etc.$) représente les exemples d'apprentissage et A_j (c.-à-d. A_b, A_c, $etc.$) représente l'ensemble des attributs utilisés pour construire l'arbre de décision correspondant. Cependant, comme il est illustré sur la figure 1, pour identifier ces ensembles d'instances et d'attributs, trois ou deux variables décisionnelles sont requises, respectivement. Pour celui de l'exemple d'apprentissage, la taille L_i des échantillons d'apprentissage définis sans répétition S_i, son identifiant id_{S_i} et l'identifiant id_{R_i} de la séquence de répétition notée R_i. Pour l'ensemble d'attributs, les deux variables sont la taille L_i de l'ensemble d'attributs A_i et son identifiant id_{A_i}. En résumé, $5 * k$ variables sont nécessaires pour représenter une forêt aléatoire composée de k arbres de décision.

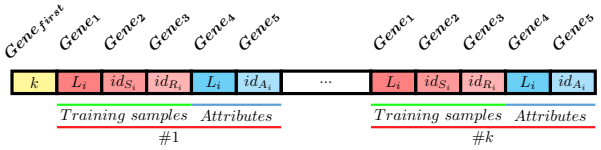


Figure 1: La représentation d'un chromosome.

Pour l'approche proposée (notée GA_RF), nous présentons un codage binaire des informations concernant les sous-ensembles représentant chacun des arbres, ainsi que la taille de la forêt. Les individus sont représentés par des tableaux de 0 et de 1 (c.-à-d. une chaîne binaire). Chaque arbre de la forêt aléatoire est codé sur le chromosome à l'aide de 5 gènes (c.-à-d. $5*k + 1$ pour représenter une forêt aléatoire de k arbres). Pour chaque solution, en utilisant ces identifiants, nous générons les ensembles d'instances et d'attributs, pour chacun des couples d'ensembles récupérés, nous construisons le bon arbre de décision suivant un modèle donné (c.-à-d., J48, BFTree, $etc.$). Une fonction de finesse sera évaluée pour tous les chromosomes à chaque itération et seules les meilleures solutions candidates sont sélectionnés pour la prochaine itération. Soit $\mathcal{G}$ un ensemble codé sur un chromosome ch, la fonction de finesse proposée est définie comme suit : $fitness(ch) = \alpha * ACC(\mathcal{G}) + \beta * DIV(\mathcal{G})$ où ACC est le taux de bonne classification du forêt aléatoire construit à base de $\mathcal{G}$, et DIV est la diversité des membres de l'ensemble lors de la classification de l'ensemble de test TS. Cette valeur consiste en la moyenne de la valeur Kappa (K) de chacun des arbres T_i avec l'ensemble des autres arbres.

3 EXPÉRIMENTATIONS

L'approche proposée pour la construction de forêts aléatoires, notée GA_RF, a été implémentée en Java en utilisant les frameworks WEKA [6] et jMetal [7], avec une population de 100 individu, une probabilité de croisement et de mutation de 50% et 1% respectivement, $\alpha = 0.8$ et $\beta = 0.2$. Nous avons évalué GA_RF sur 10 bases de données de l'UCI [5] en suivant une validation croisée 10-CV.

Le tableau 1 répertorie les résultats de GA_RF, de la forêt aléatoire standard de Weka (avec 100 arbres), ForestPA (avec 100 arbres) [9], CSForest (avec 100 arbres) [8]. Le tableau 1 indiquent claire-

Table 1: Comparaison des résultats de taux de classification de GA_RF avec un RF classique.

Base de données	[1]RF (100 DTs) AC (%) (10-CV)	[2]ForestPA (100 DTs) AC(%) (10-CV)	[3]CSForest (100 DTs) AC(%) (10-CV)	GA_RF AC(%) (10-CV)	GA_RF N° DT (10-CV)
Balance-Scale (BS)	81.76	85.12	80.80	**86.89**	72.40
Breast-Cancer (BC)	69.93	73.07	37.41	**75.22**	61.60
Credit-a (CR)	86.52	86.95	85.50	**91.59**	73.80
Dermatology (DE)	96.99	**98.36**	80.60	81.41	79.80
Diabetes (DI)	76.69	74.86	66.01	**80.85**	84.80
Ecoli (EC)	**86.30**	85.71	82.44	74.06	48.90
Heart-Statlog (HS)	83.70	82.96	70.37	**90.07**	70.60
Hepatitis (HE)	81.29	83.87	79.35	**85.12**	64.00
Iris (IR)	96.66	96.00	92.66	**98.00**	55.00
Liver-Disorders (LD)	73.04	73.33	60.28	**80.28**	62.80

ment que l'utilisation de GA_RF pour construire une forêt aléatoire conduit à une amélioration significative sur 8 jeux de données par rapport à l'approche de base RF ainsi qu'aux autres approches qui surperforment GA_RF seulement sur 2 jeux de données réduite avec une moyenne de réduction de 33% par rapport aux 100 arbres générés par les autres méthodes.

4 CONCLUSION

Dans cet article, nous avons proposé GA_RF, un algorithme pour améliorer la construction des forêts aléatoires. GA_RF cherche à combiner des processus de sélection d'attributs et d'échantillonnage d'exemples qui permettent de construire un ensemble d'arbres de décision amélioré. Les résultats expérimentaux sur 10 jeux de données ont montré que GA_RF améliore les performances par rapport à d'autres approches de la littérature.

RÉFÉRENCES

[1] KULKARNI, Vrushali Y. et SINHA, Pradeep K. Pruning of random forest classifiers : A survey and future directions. In : Data Science & Engineering (ICDSE), 2012 International Conference on. IEEE, 2012. p. 64-68.
[2] ADNAN, Md Nasim et ISLAM, Md Zahidul. Optimizing the number of trees in a decision forest to discover a subforest with high ensemble accuracy using a genetic algorithm. Knowledge-Based Systems, 2016, vol. 110, p. 86-97.
[3] BREIMAN, Leo. Random forests. Machine learning, 2001, vol. 45, no 1, p. 5-32.
[4] KARABADJI, Nour El Islem, SERIDI, Hassina, BOUSETOUANE, Fouad, et al. An evolutionary scheme for decision tree construction. Knowledge-Based Systems, 2017, vol. 119, p. 166-177.
[5] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http ://archive.ics.uci.edu/ml]. Irvine, CA : University of California, School of Information and Computer Science.
[6] HALL, Mark, FRANK, Eibe, HOLMES, Geoffrey, et al. The WEKA data mining software : an update. ACM SIGKDD explorations newsletter, 2009, vol. 11, no 1, p. 10-18.
[7] DURILLO, Juan J. et NEBRO, Antonio J. jMetal : A Java framework for multi-objective optimization. Advances in Engineering Software, 2011, vol. 42, no 10, p. 760-771.
[8] SIERS, Michael J. et ISLAM, Md Zahidul. Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem. Information Systems, 2015, vol. 51, p. 62-71.
[9] ADNAN, Md Nasim et ISLAM, Md Zahidul. Forest PA : Constructing a decision forest by penalizing attributes used in previous trees. Expert Systems with Applications, 2017, vol. 89, p. 389-403.

Jumping Evaluation of Nested Regular Path Queries

Rustam Azimov
rustam.azimov19021995@gmail.com
JetBrains Research
Saint Petersburg State University
St. Petersburg, Russia

Joachim Niehren
joachim.niehren@inria.fr
Inria Lille
Lille, France

Sylvain Salvati
sylvain.salvati@univ-lille.fr
Université de Lille
Lille, France

ABSTRACT

The propositional dynamic logic is a fundamental language that provides nested regular path queries for datagraphs, as needed for querying graph databases and RDF triple stores. We propose a new algorithm for evaluating nested regular path queries. Not only does it evaluate path queries on datagraphs from a set of start nodes in combined linear time, but also this complexity bound depends only on the size of the query's *top-down needed* subgraph, a notion that we introduce formally. For many queries relevant in practice, the top-down needed subgraph is way smaller than the whole datagraph. Our algorithm is based on a new compilation schema from nested regular path queries to monadic datalog queries that we introduce. We prove that the top-down evaluation of the datalog program visits only the top-down needed subgraph for the path query. Thereby, the combined linear time complexity depending on the size of the top-down needed subgraph is implied by a general complexity result for top-down datalog evaluation [8].

As an application, we show that our algorithm permits to reformulate in simple terms a variant of a very efficient automata-based algorithm proposed by Maneth and Nguyen that evaluates navigational path queries in datatrees based on indexes and jumping. Moreover, our variant overcomes some limitations of Maneth and Nguyen's: it is not bound to trees and applies to graphs; it is not limited to forward navigational XPath but can treat any nested regular path query and it can be implemented efficiently without any specialized or dedicated techniques, by simply using any efficient datalog evaluator.

1 INTRODUCTION

Regular path queries [6] are regular expressions for navigating in edge labeled graphs. They belong to the core of various query languages for datagraphs, as part of query languages of graph databases and RDF triple stores. Nested regular path queries (NRPQs) [4] extend on regular expressions by adding filters with logical operators, that in turn may contain the regular path queries. They were first invented as the programs of propositional dynamic logic (PDL) [3], constitute the navigational core of regular XPath where they are restricted to query datatrees, and are also part of *n*Sparql for querying knowledge stores in the semantic Web [7].

The set of nodes that can be reached by an NRPQ P on a graph G with a set of start nodes S can be computed in combined linear time, i.e. in $O(|P||G|)$. This is folklore in the context of PDL, XPath, and *n*Sparql but was first shown for the richer alternation-free modal μ-calculus [2]. However, this complexity upper bound alone is far too high in practice: if the graph is a database then it may be too large for a complete traversal for each query. Furthermore, for many queries only a fraction of the graph may be relevant for answering the query, which fraction may depend on the query answering algorithm. Therefore, we formalize a notion of needed subgraph coined as *top-down needed subgraph*, as the subgraph that is traversed with a top-down evaluation of the query. We propose a query answering algorithm with combined linear complexity *with respect to the top-down needed subgraph*, instead of the whole graph which we consider as too expensive.

For regular path queries, a canonical notion of the top-down needed subgraph seems quite intuitive. It contains all nodes and edges that are traversed when considering the path query as a description for navigation while starting in the given set of start nodes. Of course, the presence of the Kleene star makes memoization mandatory for otherwise the algorithm may loop infinitely. The part of the graph that is traversed this way is what we call the top-down needed subgraph. The notion of top-down needed nodes can then be lifted from regular path queries to NRPQs rather naturally. What becomes more tedious is to find an evaluation algorithm for NRPQs that satisfies our complexity requirement. The existing proposals in [1, 7] achieve combined linear time complexity by pre-evaluating the filters all over the graph in a bottom-up manner and then running an evaluation algorithm for regular path queries. Evaluating the filters top-down seems more difficult, since one would have to jump back to the starting node, requiring to compute a binary relation. However, the bottom-up pre-computation of the filters over all the graph may visit nodes that are *not* needed for top-down evaluation of the NRPQ so these algorithms do not satisfy the envisaged complexity bound.

As an example, consider the graph G_0 in Fig. 2 with edge labels $\{a, b, c\}$, the NRPQ $P_0 = \text{edge}_a[\text{edge}_b/\text{edge}_c]$, and set of start nodes $S_0 = \{0\}$. Query P_0 started at S_0 selects all those nodes of G_0 that are connected to the start node 0 by an a-edge, and have a path over a b-edge followed by a c-edge. The top-down algorithm with pre-evaluation of filters will first compute the answer set of the filter $[\text{edge}_b/\text{edge}_c]$, which is $\{1, 4, 5\}$. It will then compute the set of nodes that are reached from the start node 0 over an a-edge, which is $\{1, 4, 6\}$. The answer set is the intersection, which is $\{1, 4\}$. This algorithm, however, will inspect some nodes and edges for the pre-evaluation of the filters that are *not* top-down needed, namely the node 5 and the b-edge from 5 to 2.

We will show that this complexity problem can be avoided by enhancing the naive top-down evaluator with memoization – instead of precomputing the filter queries. The right kind of memoization can be obtained by compiling the path query into a monadic datalog

$$q_0(x) :- q_1(x), q_2(x).$$
$$q_1(y) :- \mathit{start}(x), \mathrm{edge}_a(x, y).$$
$$q_2(x) :- \mathrm{edge}_b(x, y), q_3(y).$$
$$q_3(y) :- \mathrm{edge}_c(y, z).$$

Figure 1: The Datalog program M_0 for the nested regular path query $P_0 = \mathrm{edge}_a[\mathrm{edge}_b/\mathrm{edge}_c]$.

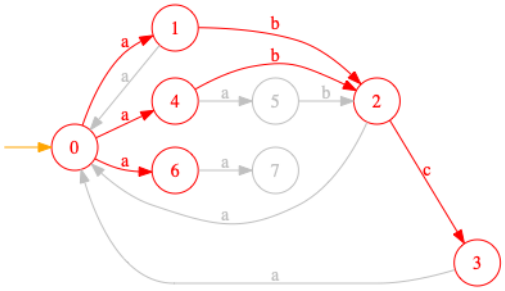


Figure 2: Graph G_0, start set $S_0 = \{0\}$, and the top-down needed subgraph for P_0 in red.

program, and then evaluating this datalog program in a top-down manner. Even though monadic, the datalog program may still use extensional predicates of higher arities. In the case of P_0 we obtain the datalog program in Fig. 1, which for talking about graphs with edge labels in $\{a, b, c\}$ uses the binary extensional predicates $\mathrm{edge}_a, \mathrm{edge}_b, \mathrm{edge}_c$. Furthermore, there is the monadic extensional predicate *start* for representing the start set. We note that a filter query such as $[\mathrm{edge}_b/edge_c]$ is compiled quite differently (see the rules of the intensional predicates q_2 and q_3) to how one would compile the path query $\mathrm{edge}_b/edge_c$. The reason is that a filter query returns the node where the path starts – under the condition that some node is reached at the end – while the path query selects all the nodes reached at the end.

Our first contribution is an algorithm that answers NRPQs in the combined linear time $O(|tdn_{G,S}(P)||P|)$ with respect to the size of top-down needed subgraph $tdn_{G,S}(P)$. For this, we present a linear time compilation scheme for mapping path queries to datalog queries. For the sake of presentation, we treat only negation-free NRPQs, so that stratified negation is not needed. We prove that the compiler is correct in that if it transforms a query P and a start set S into a datalog query M, then top-down needed subgraph $tdn_{G,S}(P)$ is the part of the graph's database that is visited by top-down evaluation of the datalog query M on the database. Furthermore, the datalog queries produced are monadic, and restricted in such a way that their top-down evaluation can be done in combined linear time depending on the size of the top-down visited subdatabase (Ullman's Theorem 3 on p9 of [9]) and [8] for an extensions with stratified negation). It follows that the answer set of the NRPQ on the graph with start set S can indeed be computed in time $O(|tdn_{G,S}(P)||P|)$.

Our algorithm can be extended to a jumping algorithm for answering NRPQs on graphs with indexes. The indexes are binary relations defined by other NRPQs that allow the algorithm to jump in the graph. For instance, when given an index for the NRPQ $I = \mathrm{edge}^*/a$? on the input graph, the evaluation algorithm can always jump to all a-labeled nodes accessible from the current node, without visiting the intermediates. We consider that the indexes are given with the input, since they are usually pre-computed elsewhere. Therefore, the indexes can simply by integrated into the graph as new edges that are labeled by the index's name, which is I in our example. Furthermore, the NRPQ is then rewritten by substituting all occurrences of I as a subquery in the NRPQ by edge_I, so that we can apply the previous machinery. An efficient implementation of our algorithm can be based on any efficient top-down datalog evaluator, since it is sufficient to evaluate the monadic datalog program produced by our compiler.

As an application our jumping algorithm permits to reformulate in simple terms a very efficient automata-based algorithm proposed by Maneth and Nguyen [5] that evaluates NRPQs on datatrees with indexes based on jumping. More precisely, their algorithm covers navigational forwards XPath queries on XML documents. It is based on alternating tree automata with selection states (which can be seen as binary datalog programs while ours are monadic). Our approach overcomes the limitations of theirs: it is not bound to trees but applies to graphs; it is not limited to forward navigational XPath but can treat any NRPQs also with backward steps, and it can be implemented efficiently without any specialized or dedicated techniques.

ACKNOWLEDGMENTS

The research was supported by the Russian Science Foundation grant 18-11-00100.

REFERENCES

[1] Marcelo Arenas and Jorge Pérez. 2011. Querying semantic web data with SPARQL. In *Proceedings of the thirtieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 305–316.

[2] Rance Cleaveland and Bernhard Steffen. 1991. A Linear-Time Model-Checking Algorithm for the Alternation-Free Modal Mu-Calculus. In *Proceedings of the 3rd International Workshop on Computer Aided Verification (CAV '91)*. Springer-Verlag, Berlin, Heidelberg, 48–58.

[3] Michael J. Fischer and Richard E. Ladner. 1979. Propositional Dynamic Logic of Regular Programs. *J. Comput. Syst. Sci.* 18, 2 (1979), 194–211. https://doi.org/10.1016/0022-0000(79)90046-1

[4] Leonid Libkin, Wim Martens, and Domagoj Vrgovc. 2013. Querying Graph Databases with XPath. In *Proceedings of the 16th International Conference on Database Theory* (Genoa, Italy) *(ICDT '13)*. Association for Computing Machinery, New York, NY, USA, 129–140. https://doi.org/10.1145/2448496.2448513

[5] Sebastian Maneth and Kim Nguyen. 2010. XPath Whole Query Optimization. *PVLDB* 3, 1 (2010), 882–893. http://www.comp.nus.edu.sg/~vldb2010/proceedings/files/papers/R79.pdf

[6] Wim Martens and Tina Trautner. 2018. Evaluation and Enumeration Problems for Regular Path Queries. In *21st International Conference on Database Theory (ICDT 2018) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 98)*, Benny Kimelfeld and Yael Amsterdamer (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 19:1–19:21. https://doi.org/10.4230/LIPIcs.ICDT.2018.19

[7] Jorge Pérez, Marcelo Arenas, and Claudio Gutiérrez. 2010. nSPARQL: A navigational language for RDF. *J. Web Semant.* 8, 4 (2010), 255–270. https://doi.org/10.1016/j.websem.2010.01.002

[8] K. Tekle and Yanhong Liu. 2010. Precise complexity analysis for efficient Datalog queries. In *PPDP'10 - Proceedings of the 2010 Symposium on Principles and Practice of Declarative Programming*. 35–44. https://doi.org/10.1145/1836089.1836094

[9] J. D. Ullman. 1989. Bottom-up Beats Top-down for Datalog. In *Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (Philadelphia, Pennsylvania, USA) *(PODS '89)*. Association for Computing Machinery, New York, NY, USA, 140–149. https://doi.org/10.1145/73721.73736

Schema Inference for Property Graph Databases

Hanâ Lbath
Lyon 1 University, CNRS Liris
UdL, CNRS, ENS Lyon, UCBL1
France
hana.lbath@ens-lyon.fr

Angela Bonifati
Lyon 1 University, CNRS Liris
France
angela.bonifati@univ-lyon1.fr

Russ Harmer
UdL, CNRS, ENS Lyon, UCBL1
France
russ.harmer@ens-lyon.fr

ABSTRACT

Property graph instances are typically built without necessarily predefining a schema. Although this ensures great flexibility, this can also become a great impediment, notably whenever the structure of the underlying instances stabilizes. Since several graph instances exist prior to the schema definition, extracting the schema from those instances in a principled way might become a daunting task. In this paper, we present an end-to-end schema inference method for property graph schemas that tackles complex and nested property values, multi-labeled nodes and node hierarchies. Our method consists of three steps, the first of which builds upon Cypher queries to extract the node and edge serialization of a property graph. The second step builds over a MapReduce type inference system, working on the serialized output thereby obtained during the first step. The third step analyzes subtypes and supertypes to infer node hierarchies. We describe our schema inference pipeline and its implementation, a labels- and a properties-oriented variant. Finally, we experimentally evaluate and compare the scalability and accuracy of our approaches on several real-life datasets.

Graph-based keyword search in heterogeneous data sources

Angelos Christos Anadiotis
angelos.anadiotis@polytechnique.edu
Ecole Polytechnique, Institut Polytechnique de Paris and EPFL
Palaiseau, France

Mhd Yamen Haddad
mhd-yamen.haddad@inria.fr
Inria Saclay and Institut Polytechnique de Paris
Palaiseau, France

Ioana Manolescu
ioana.manolescu@inria.fr
Inria Saclay and Institut Polytechnique de Paris
Palaiseau, France

ABSTRACT

Data journalism is the field of investigative journalism which focuses on digital data by treating them as first-class citizens. Following the trends in human activity, which leaves strong digital traces, data journalism becomes increasingly important. However, as the number and the diversity of data sources increase, heterogeneous data models with different structure, or even no structure at all, need to be considered in query answering.

Inspired by our collaboration with Le Monde, a leading French newspaper, we designed a novel query algorithm for exploiting such heterogeneous corpora through keyword search. We model our underlying data as graphs and, given a set of search terms, our algorithm finds links between them within and across the heterogeneous datasets included in the graph. We draw inspiration from prior work on keyword search in structured and unstructured data, which we extend with the data heterogeneity dimension, which makes the keyword search problem computationally harder. We implement our algorithm and we evaluate its performance using synthetic and real-world datasets.

Our full-length paper is available at:
https://hal.inria.fr/hal-02934277

The ConnectionLens system is available online at:
https://gitlab.inria.fr/cedar/connectionlens

Optimization for Large-Scale Fuzzy Joins Using Fuzzy Filters in MapReduce

Thi-To-Quyen TRAN
Univ Rennes, CNRS, IRISA
Lannion, France
thi-to-quyen.tran@irisa.fr

Thuong-Cang PHAN
Cantho University
Cantho, Vietnam
ptcang@cit.ctu.edu.vn

Anne LAURENT
Univ Montpellier, LIRMM, CNRS
Monpellier, France
Anne.Laurent@lirmm.fr

Laurent d'Orazio
Univ Rennes, CNRS, IRISA
Lannion, France
laurent.dorazio@univ-rennes1.fr

ABSTRACT

A fuzzy or similarity join is one of the most useful data processing and analysis operations for Big Data in a general context. It combines pairs of tuples for which the distance is lower than or equal to a given threshold ε. The fuzzy join is used in many practical applications, but it is extremely costly in time and space, and may even not be executed on large-scale datasets. Although there have been some studies to improve its performance by applying filters, a solution of an effective fuzzy filter for the join has never been conducted. In this paper, we thus extend our previous work by proposing a novel fuzzy filter to optimize fuzzy joins. This filter is a compact, probabilistic data structure that supports very fast similarity queries by maintaining a bit matrix, with small false positive rate and zero false negative rate. We show that our proposal is more efficient than others because of eliminating redundant data, reducing computation cost and avoiding duplicate output.

KEYWORDS

Fuzzy join, Similarity join, MapReduce, Fuzzy Filter

1 INTRODUCTION

One challenge in distributed computing is to avoid expensive data transmission and large disk I/Os. While recent studies on the fuzzy join [1, 3, 6, 8] have common limitations such as input re-reading by multiple phases, wasteful redundancy and duplication of data, we have introduced filter-based approaches [4, 5, 7]. Our team was interested in using Bloom Filters [2], Intersection Filter [5]. The idea is to filter irrelevant data as soon as possible to reduce data transfers and workload on different machines. Besides, the computation of the Hamming distance is shown faster than the computation of the distance in the input space. Therefore, we take advantage of its theory to propose a new filter for fuzzy joins.

2 FUZZY FILTER STRUCTURE

The idea begins with finding the intersection between dataset and balls. The Fuzzy filter $FF(S)$ combines a Bloom filter $BF(S)$ to identify elements, and a table to store real similar elements in the ball of each element. The m bit Bloom filter $BF(S)$ uses one hash function h to calculate positions for an element of S and sets the bit at the resulting positions. The ball list is an array of size m of m bit Bloom Filters (a matrix of $m \times m$ bits), each one stores its ball $BF(B(s_i))$.

The build operation of the fuzzy filter $FF(S)$ is described as follows.

- (0) Hash each ball to a bit array of length m. With b-bit string, it exists 2^b balls, each ball has about $b^r/r!$ elements. Let us recall the assumption that hash operation performs in unit time. This step has the cost $C_{(0)} \approx 2^b b^r/r!$
- (1) Hash S to bit array of length m. $C_{(1)} = |S|$
- (2) Ball list compute by the intersection of $BF(S)$ and $BF(B(s_i))$. $C_{(2)} = |2^b|$
- The build cost for $FF(S)$ is

$$C_{FF(S)-build} \approx \frac{b^r}{r!}2^b + |S| + 2^b = (\frac{b^r}{r!} + 1)2^b + |S|$$

3 OPTIMIZING LARGE-SCALE FUZZY JOINS

With the integration of FF, our proposal ignores the costly and redundant ball calculation. Specifically, the FF-FJ algorithm consists of two phases:

- Stage 1 (Pre-processing): A filter $FF(S)$ is built on a join key set of the input dataset S. Each worker hashes tuples of input splits to find $h(s_i)$, emits a list of $[h(s_i)]$ to one reducer for the FF building. Thus, the Map cost is $M = |S|$, the communication cost is $D = \#mappers$, the computation cost is the FF building cost $C_{FF(S)-build} \approx \frac{b^d}{d!}2^b + 2^b = (\frac{b^d}{d!} + 1)2^b$. If the ball list is pre-known, the processing cost is only 2^b of AND operations. Figure **??** describes an example of the pre-processing stage of FF-FJ for the fuzzy join.
- Stage 2 (Join processing): $FF(S)$ is distributed to all the computing nodes and is used to quickly emit real similar elements of the input dataset during the map phase. Each record s is hashed by $h(s)$. $BF[h(s)]$ returns a list of similar elements. Finally, mappers emit the intermediate tuple with the key is in

form $(h(s), h(t_i))$ if $h(s) < h(t_i)$ and vice versa. The number of reducers is $(2^b)(2^b - 1)/2$

$$s \xrightarrow[FF(S)]{map} \begin{cases} ((h(s), h(t_i)), s), & \forall t_i \in B_r(s) \cap S, h(s) < h(t_i) \\ ((h(t_i), h(s)), s), & \forall t_i \in B_r(s) \cap S, h(t_i) < h(s) \end{cases}$$

The map cost for each record in this phase is only 1, instead of $k|B_r|$ in BF-BH algorithm. The number of intermediate elements for each record is optimized, instead of $|B_r|$ in BH algorithm, and may be approximate with BF-BH algorithm because of the same filtering technique.

In the ideal case, regardless of the false positive, our approach has no redundant intermediate data and no duplicated results without verification in reducers.

4 FF ANALYSIS AND OPTIMIZATION

With an overview structure as above, FF can be applied to some data types (string, vector, set), some distance functions (Hamming, edit distance) as long as the balls can be calculated. In the binary space, FF uses $m = 2^b$, the exact probabilities of filtering are guaranteed 100%, without false probabilities. However, in practice, for a large finite alphabet set, a large string length, to optimize memory, the filter size is designed to be smaller than the actual set size. Hence, it may lead to a false probability.

In the case of multiple balls that have the same hash index position, the ball in this position is the union of these collision balls. The response will include the real similar elements and also the mistaken records in another collision ball. These records are mistakenly assumed to be a similar element and must be calculated the distance in the join step.

The small false positive probability is caused by one of two cases follows

- (1) for the filter: an element in another collision ball is returned as an answer.
- (2) for the join step: an irrelevant record of S has the same hash index with an exact answer.

Conversely, it does not exist a false negative probability. In other words, no real similar element is not answered in the response.

The precision of the fuzzy filter depends on the similarity function complexity, the size of FF (m), the quality of the hash function. For example, with Hamming distance threshold r, a dataset S of b bit-strings, for n real elements ($|S| = n < 2^b$), the false positive of a hash bucket list of size m bits is

$$f_S = 1 - (1 - 1/m)^n$$

Each hash ball contains about $b^r/r!$ elements. So its number of possible collision bits is approximate $(b^r/r!)(1 - 1/m)^{b^r/r!}$. If a collision occurs, the probability of a bit 1 is out of the real ball is

$$\frac{b^r(1 - 1/m)^{b^r/r!}}{mr!}(1 - \frac{b^r(1 - 1/m)^{b^r/r!}}{mr!})$$

However, this false bit becomes a false positive answer only if its index in hash bucket S is also set.

The building of the matrix that includes all the balls with a reasonable alphabet, an acceptable length of a string is feasible. For large-scale datasets, avoiding repeated calculations for the pre-known balls will reduce a large workload. In cases where the set of balls cannot be pre-calculated, the input dataset S can be read one time as a distinct key set to compute its balls. This cost can be amortized, especially using streaming or caching techniques (e.g Spark [9]).

Another advantage of FF is the flexibility with distance, capable of equi-join and fuzzy join. The ball list can be easily updated quickly as soon as a new record appears. This can be applied in stream join applications. The solution given is that the AND operation in step (3) during the building phase is not performed. The ball list stores all the balls. The answer to each new query t is the intersection of S and the ball $B(t)$.

5 CONCLUSION

In this paper, we study theoretical details on large-scale fuzzy join algorithms in MapReduce. We propose approaches for building a Fuzzy Filter, a scalable solution with respect to the distance and the volume of the input datasets. This filter is a compact, probabilistic data structure that supports very fast similarity queries by maintaining a bit matrix, with small false positive rate and zero false negative rate. We show the relevance of this structure in fuzzy self join. In addition, our solution for the FF-FJ algorithm is more efficient than previous solutions without filters or with a Bloom Filter since it significantly reduces redundant data, costly and wasteful computations, and thus produces fewer intermediate data, eliminates duplicated results, and avoids unnecessary comparisons. Although FF-FJ algorithm has false positives and an extra cost for the pre-processing step, its efficiency in space-saving and filtering often outweighs these drawbacks. We use the MapReduce cost model to prove it.

Future work includes extending Intersection Fuzzy Filter for fuzzy two-way join, fuzzy multiway join and fuzzy recursive join. Our optimizations may be extended in the cache or streaming supported framework to reuse the pre-processing cost. Perspectives also include validating our solutions, comparing them with other approaches and extending the research for other fuzzy join algorithms. Besides, experimental evaluation and a solution for skewness problems will also be considered.

REFERENCES

[1] Foto N. Afrati, Anish Das Sarma, David Menestrina, Aditya Parameswaran, and Jeffrey D. Ullman. 2012. Fuzzy Joins Using MapReduce. In *ICDE*. 498–509.

[2] Burton H. Bloom. 1970. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM* 13, 7 (1970), 422–426.

[3] D. Deng, G. Li, S. Hao, J. Wang, and J. Feng. 2014. MassJoin: A mapreduce-based method for scalable string similarity joins. In *2014 IEEE 30th International Conference on Data Engineering*. 340–351.

[4] Thuong-Cang Phan, Laurent d'Orazio, and Philippe Rigaux. 2016. A Theoretical and Experimental Comparison of Filter-Based Equijoins in MapReduce. *TLDKS* 25 (2016), 33–70.

[5] Thuong-Cang Phan, Laurent d'Orazio, and Philippe Rigaux. 2013. Toward Intersection Filter-based Optimization for Joins in MapReduce. In *Cloud-I*. 2:1–2:2.

[6] Chuitian Rong, Chunbin Lin, Yasin Silva, Jianguo Wang, Wei Lu, and Xiaoyong Du. 2017. Fast and Scalable Distributed Set Similarity Joins for Big Data Analytics. 1059–1070.

[7] Thi-To-Quyen Tran, Thuong-Cang Phan, Anne Laurent, and Laurent d'Orazio. 2018. Improving Hamming distance-based fuzzy join in MapReduce using Bloom Filters. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 1–7.

[8] Rares Vernica, Michael J. Carey, and Chen Li. 2010. Efficient Parallel Set-similarity Joins Using MapReduce. In *SIGMOD*. 495–506.

[9] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*. USENIX Association, 10.

Graph integration of structured, semistructured and unstructured data for data journalism

Oana Balalau
oana.balalau@inria.fr
Inria Saclay and Institut Polytechnique de Paris
Palaiseau, France

Catarina Conceição
catarina.conceicao@tecnico.ulisboa.pt
INESC-ID and IST, Univ. Lisboa, Portugal
Porto Salvo, Portugal

Helena Galhardas
catarina.conceicao@tecnico.ulisboa.pt
INESC-ID and IST, Univ. Lisboa, Portugal
Porto Salvo, Portugal

Ioana Manolescu
ioana.manolescu@inria.fr
Inria Saclay and Institut Polytechnique de Paris
Palaiseau, France

Tayeb Merabti
tayeb.merabti@inria.fr
Inria Saclay and Institut Polytechnique de Paris
Palaiseau, France

Youssr Youssef
tayeb.merabti@inria.fr
Inria Saclay and Institut Polytechnique de Paris
Palaiseau, France

Jingmao You
tayeb.merabti@inria.fr
Inria Saclay and Institut Polytechnique de Paris
Palaiseau, France

ABSTRACT

Nowadays, journalism is facilitated by the existence of large amounts of digital data sources, including many Open Data ones. Such data sources are extremely heterogeneous, ranging from highly structured (relational databases), semi-structured (JSON, XML, HTML), graphs (e.g., RDF), and text. Journalists (and other classes of users lacking advanced IT expertise, such as most non-governmental-organizations, or small public administrations) need to be able to make sense of such heterogeneous corpora, even if they lack the ability to define and deploy custom extract-transform-load workflows. These are difficult to set up not only for arbitrary heterogeneous inputs, but also given that users may want to add (or remove) datasets to (from) the corpus.

We describe a complete approach for integrating dynamic sets of heterogeneous data sources along the lines described above: the challenges we faced to make such graphs useful, allow their integration to scale, and the solutions we proposed for these problems. Our approach is implemented within the ConnectionLens system; we validate it through a set of experiments.

Our full-length paper is available at:
https://hal.inria.fr/hal-02904797

The ConnectionLens system is available online at:
https://gitlab.inria.fr/cedar/connectionlens

An extension of chronicles temporal model with taxonomies - Application to epidemiological studies

Johanne Bakalara
Univ Rennes, EA-7449 REPERES and Univ Rennes, Inria, IRISA-UMR6074
France

Thomas Guyet
Institut Agro, IRISA-UMR6074
France

Olivier Dameron
Univ Rennes, Inria, IRISA-UMR6074
France

André Happe
CHRU Brest
France

Emmanuel Oger
Univ Rennes, EA-7449 REPERES
France

KEYWORDS

Temporal query, Medico-administrative databases, Sequences of events, Chronicles, Semantic Web.

1 INTRODUCTION

Pharmaco-epidemiology (PE) studies the conditions and consequences of health products, *i.e.* drugs or medical devices usage at the population scale in real situations using methodologies developed in general epidemiology.

Modern PE relies on administrative databases to perform such studies on care trajectories, *i.e.* on patient-centered sequences of drugs deliveries, medical procedures and hospitalisations. The use of medico-administrative databases (MADB) is useful in PE studies, since data are readily available and cover a large population. The problem with MADB is the semantic gap between raw data and the epidemiological question.

On the one hand, epidemiologists are looking for medical events. For instance, they would like to identify patients suffering from *venous thromboembolism* (VTE). On the other hand, raw data are related to reimbursements of medical acts or drug deliveries. There is no exploitable diagnosis available in administrative databases and no clinical results related to medical acts or exams. For instance, a patient having a lower limbs doppler ultrasonography exam and few days after a delivery of anticoagulant drugs for 3 to 6 or 12 months is probably suffering from VTE. As MADB record medical exams and drugs deliveries, the above description may be used as a proxy of VTE.

The challenge for epidemiologists is to define phenotypes of medical events [3], *i.e.* a combination of information available in the database that reveals an occurrence of a medical event. This article addresses the problem of enumerating the occurrences of a complex temporal pattern (*i.e.* phenotypes) in a dataset of care trajectories. Our contribution is threefold: (i) we formalise the task of finding patients in a MADB verifying a phenotype; (ii) we developed a tool HyCOR to perform the task. It's an hybrid method combining the expressiveness of Semantic Web and the efficiency of a pattern occurrence enumeration algorithm; (iii) we evaluate HyCOR on a real case study of enumerating VTE events in the French MADB.

2 FORMALISATION – SEQUENCES AND CHRONICLES

Firstly, this paper proposes a formalisation of the data contained in the MADB and a generic temporal model to express phenotypes of medical events.

Sequences. MADB is seen as a longitudinal view where each patient is represented by a sequence of timestamped cares, so called *events*. Formally, an *event* is a pair (e, t) where e is an event label and $t \in \mathbb{N}$ is a timestamp (in days).

Chronicles. A *chronicle* is a multiset of event labels and a set of temporal constraints between couples of events [2]. In our applied context, chronicle enables to represent a phenotype [1]. This paper proposes a chronicle extension where event may have label belonging to the equivalence class of an event label in a given taxonomy.

Chronicles enumeration recognition. The enumeration of chronicles' occurrences aims at localizing where a medical pattern occurs in a patient care trajectory and so on, to find every sequences verifying a chronicle.

Example. Patients are stored in the database as sequences. Table 1 shows some examples of sequence. Each sequence is made of drug deliveries events (couples of label and timestamp). Labels belongs to ATC codes, *i.e.* the code of a delivered drug in the ATC toxonomy. The chronicle in Figure 2 represents a phenotype used to find patients exposed for at least 3 months to anticoagulant (B01A code in ATC taxonomy referees to any anticoagulant delivery). It states that an event with a label in the equivalence class of B01A is followed by an event B01A within a delay of $[1, 31]$ units of time (*ut*). The later is followed by an event B01A within a delay of $[1, 31]$ *ut*. Note that temporal constraints (intervals) between different items can contains negative numbers.

Sequences s_1 and s_2 are recognised because there are at least 3 occurrences of drug deliveries which are subclasses of B01A and temporal constraints are respected at least once. s_3 is not recognised because there is no set of events satisfying the temporal constraints.

3 SEMANTIC WEB AND HYCOR – HYBRID CHRONICLE OCCURRENCE RECOGNITION

Semantic Web is suitable to store sequences and to encode chronicle enumeration with SPARQL. So, we propose to construct in RDF a

id	Sequence
s_1	(A01AA01, 1), (B01AA01, 3), (A01AB14, 10), (B01AB02, 30), (B01AA01, 60), (B01AA01, 97)
s_2	(B01AA02, 30), (B01AA02, 40), (A01AA01, 45), (B01AB01, 47)
s_3	(A03AA01, 1), (B01AA01, 4), (C01AA01, 5), (B01AA01, 6), (B01AA01, 40), (D01AA01, 9)

Figure 1: Example of a dataset of three sequences (three patients)

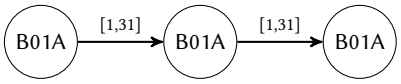


Figure 2: Chronicle for representing anticoagulant exposition

database made of sequences and to encode a chronicle enumeration in SPARQL. We propose two approaches for chronicle enumeration: the first approach fully uses SPARQL and the second approach is an hybrid tool combining SPARQL query and a dedicated algorithm.

full-SPARQL. The full-SPARQL approach constructs one SPARQL-query to find all the sequences verifying the chronicle. SPARQL is expressive enough for representing chronicles. However the queyring time is long. A SPARQL query can not compete with dedicated enumeration algorithms as its solver strategy is not optimised for this task.

Therefore, we propose an hybrid approach to benefit from the best of both fields: efficiency of dedicated approaches and expressiveness of Semantic Web.

HyCOR. The HyCOR process works in two steps. First, a SPARQL query yields flattened sequences. A flattened sequence contains only the sequence events that belong to equivalent class of chronicle event labels, and represented by chronicle labels themselves. Second, HyCOR applies a dedicated enumeration algorithms to enumerate chronicle occurrences in the flattened sequences.

4 EXPERIMENTS

Several synthetic datasets have been generated. Each dataset contains a set of sequences where event labels are randomly chosen at the lowest level of ATC taxonomy. The ATC taxonomy contains 1 900 classes. In addition, occurrences of ten 15-sized chronicles are embedded in the dataset. For each chronicle, a constraint is generated for each pair of events. The synthetic dataset generation process ensures that each chronicle occurs in about 20% of the sequences. Experiments evaluate the impact of two main parameters on execution times of SPARQL and HyCOR: the size of the dataset (number of sequences and number of events per sequence) and the chronicle size. It shows that HyCOR is at least one order of magnitude more efficient than pure SPARQL. Experiments show that SPARQL does not scale up for datasets containing more than 15 000 sequences.

We also evaluate the part of HyCOR execution times spent by the SPARQL mapping and the chronicle enumeration algorithm. On average, the SPARQL query execution represents 85% ± 3.47 of the total execution time

5 USE CASE ON THE SNDS TO FIND PATIENTS WITH THROMBOEMBOLISM

Our use case proposes to find patients diagnosed with *venous thromboembolism* (VTE) in the SNDS. The SNDS is the french national health insurance database, which covers most of the french population (above 65 million inhabitants). The advantage of this database is to gather information about most the reimbursed medical events, from drug deliveries to nurse home cares, specialist consultations, etc. The range of medical events that are recorded in the database makes it suitable to conduct a wide variety of health studies [4]. However, SNDS has been designed for administrative purposes (care reimbursements) and does not contain detailed medical information such as medical reports, laboratory results or diagnosis.

This use case uses a geographical-based SNDS subset (the north western French Brittany population) which contains 377 359 individuals. We described VTE with the chronicle Figure 3.

`ccam:thrombose` is an union of 36 different codes of medical acts practiced in case of VTE suspicion. The codes are issued of the french taxonomy of medical acts (CCAM).

HyCOR finds 2 568 patients having VTE in 56.21*s*. (of which 52.86*s* in the SPARQL query execution),

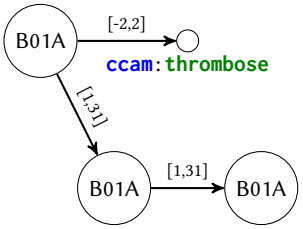


Figure 3: Chronicle to represent patients suffering of VTE

REFERENCES

[1] Yann Dauxais, Thomas Guyet, David Gross-Amblard, and André Happe. 2017. Discriminant chronicles mining. In *Proc. of Conf. on Artificial Intelligence in Medicine in Europe (AIME)*. 234–244.

[2] Christophe Dousson and Pierre Le Maigat. 2007. Chronicle Recognition Improvement Using Temporal Focusing and Hierarchization.. In *Proc. of Int. Join Conf. on Artificial Intelligence (IJCAI)*. 324–329.

[3] Na Hong, Andrew Wen, Daniel J Stone, Shintaro Tsuji, Paul R Kingsbury, Luke V Rasmussen, Jennifer A Pacheco, Prakash Adekkanattu, Fei Wang, Yuan Luo, et al. 2019. Developing a FHIR-based EHR phenotyping framework: A case study for identification of patients with obesity and multiple comorbidities from discharge summaries. *J. of Biomedical Informatics* 99 (2019), 103310.

[4] P Tuppin, J Rudant, P Constantinou, C Gastaldi-Ménager, et al. 2017. Value of a national administrative database to guide public decisions: From the système national d'information interrégimes de l'Assurance Maladie (SNIIRAM) to the système national des données de santé (SNDS) in France. *Revue d'épidémiologie et de santé publique* 65 (2017).

Selectivity correction with online machine learning

Max Halford
maxhalford25@gmail.com
IRIT Laboratory
IMT Laboratory

Philippe Saint-Pierre
philippe.saint-pierre@math.univ-toulouse.fr
IMT Laboratory

Franck Morvan
frank.morvan@irit.fr
IRIT Laboratory

ABSTRACT

Computer systems are full of heuristic rules which drive the decisions they make. These rules of thumb are designed to work well on average, but ignore specific information about the available context, and are thus sub-optimal. The emerging field of machine learning for systems attempts to learn decision rules with machine learning algorithms. In the database community, many recent proposals have been made to improve selectivity estimation with batch machine learning methods. Such methods are all batch methods which require retraining and cannot handle concept drift, such as workload changes and schema modifications. We present online machine learning as an alternative approach. Online models learn on the fly and do not require storing data, they are more lightweight than batch models, and finally may adapt to concept drift. As an experiment, we teach models to improve the selectivity estimates made by PostgreSQL's cost model. Our experiments make the case that simple online models are able to compete with a recently proposed deep learning method.

CCS CONCEPTS

• **Information systems** → **Query optimization**; • **Computing methodologies** → **Online learning settings**.

KEYWORDS

query optimisation, selectivity estimation, online machine learning, concept drift

Not Elimination and Witness Generation for JSON Schema (short version)

Mohamed-Amine Baazizi
Sorbonne Université, LIP6 UMR 7606
baazizi@ia.lip6.fr

Dario Colazzo
Université Paris-Dauphine, PSL Research University
dario.colazzo@dauphine.fr

Giorgio Ghelli
Dipartimento di Informatica, Università di Pisa
ghelli@di.unipi.it

Carlo Sartiani
DIMIE, Università della Basilicata
carlo.sartiani@unibas.it

Stefanie Scherzinger
Universität Passau
stefanie.scherzinger@uni-passau.de

ABSTRACT

JSON Schema is an evolving standard for the description of families of JSON documents. JSON Schema is a logical language, based on a set of *assertions* that describe features of the JSON value under analysis and on logical or structural combinators for these assertions. As for any logical language, problems like satisfaction, not-elimination, schema satisfiability, schema inclusion and equivalence, as well as witness generation, have both theoretical and practical interest. While satisfaction is trivial, all other problems are quite difficult, due to the combined presence of negation, recursion, and complex assertions in JSON Schema. To make things even more complex and interesting, JSON Schema is not algebraic, since we have both syntactic and semantic interactions between different keywords in the same schema object.

With such motivations, we present in this paper an algebraic characterization of JSON Schema, obtained by adding opportune operators, and by mirroring existing ones. We present then algebra-based approaches for dealing with not-elimination and witness generation problems, which play a central role as they lead to solutions for the other mentioned complex problems.

KEYWORDS

JSON Schema, negation, witness generation

1 INTRODUCTION

JSON Schema [2] is an evolving standard for the description of families of JSON documents. It is maintained by the Internet Engineering Task Force IETF [1]. Its latest version has been produced on 2019-09 [9] but is not widely used compared to the intermediate Draft-06.

JSON Schema uses the JSON syntax. Each construct is defined using a JSON object with a set of fields describing assertions relevant for the values being described. Some assertions can be applied to any JSON value type (e.g., *type*), while others are more specific (e.g., *multipleOf* that applies to numeric values only). The syntax and semantics of JSON Schema have been formalized in [8] following the specification of Draft-04. We limit ourself to an informal discussion revealing the possible constraints associated to each kind of type:

- when defining a *string*, it is possible to restrict its length by specifying the *minLength* and *maxLength* constraints and to define the *pattern* that the string should match;
- when defining a *number*, it is possible to define its range of values by specifying any combination of *minimum* / *exclusiveMinimum* and *maximum* / *exclusiveMaximum*, and to define whether it should be *multipleOf* a given number;
- when defining an *object*, it is possible to define its *properties*, the type of its *additionalProperties* and the type of the properties matching a given pattern (i.e. *patternProperties*). It is also possible to restrict the minimum and maximum number of properties using *minProperties* and *maxProperties*, and to indicate which properties are *required*;
- when defining an *array*, it is possible to define the type of its *items* and the type of the *additionalItems* which were not already defined by *items*, and to restrict the minimum and maximum size of the array; moreover, it is also possible to enforce unicity of the items using *uniqueItems*.

JSON Schema is a logical language allowing for combining assertions using standard boolean connectives: *not* for negation, *allOf* for conjunction, *anyOf* for disjunction, and *oneOf* for exclusive disjunction. As for any logical language, the following problems have a theoretical and practical interest:

- satisfaction $J \vDash S$: does a JSON document J satisfy schema S?
- not-elimination: is it possible to rewrite a schema to an equivalent form without negation?
- satisfiability of a schema: does a document J exist such that $J \vDash S$?
- schema inclusion $S \subseteq S'$: does, for each document J, $J \vDash S \Rightarrow J \vDash S'$?
- schema equivalence $S \equiv S'$: does, for each document J, $J \vDash S \Leftrightarrow J \vDash S'$?
- witness generation: is there an algorithm to generate one element J for any non-empty schema S?

While satisfaction is trivial, all other problems are quite difficult, due to the combined presence of negation, recursion, and complex assertions.

A second aspect that makes the task difficult is the non-algebraic nature of JSON Schema. A language is "algebraic" when the applicability and the semantics of its operators only depends on the

semantics of their operands. In this sense, JSON Schema is not algebraic, since we have both syntactic and semantic interactions between different keywords in the same schema object, such as the prohibition to repeat a keyword inside a schema object, or the interactions between the "`properties`" and "`additionalProperties`" keywords. For instance, the following schema[1] demands that any properties other than `foo` and `bar` must have boolean values.

```
{ "properties": {"foo": {}, "bar": {}},
  "additionalProperties": {"type": "boolean"} }
```

Such features complicate the tasks of reasoning about the language and of writing code for its manipulation.

2 SUMMARY OF CONTRIBUTIONS

JSON Algebra. We define a *core algebra*, which features a subset of JSON Schema assertions. This algebra is minimal, that is, no operator can be defined starting from the others.

Not elimination. We show that negation cannot be eliminated from JSON Schema, since there are some assertions whose complement cannot be expressed without negation such as `uniqueItems` or `multipleOf`. We enrich the core algebra with primitive operators to express those missing complementary operators, and we present a *not elimination* algorithm for the enriched algebra. To our knowledge, this is the first paper where not elimination is completely defined, with particular regard to the treatment of negation and recursion.

Witness generation. We define an approach for witness generation for the complete JSON Schema language, with the only exception of the `uniqueItems` operator, hence solving the satisfiability and inclusion problems for this sublanguage.

For space reasons, many details and formal aspects presented in the complete report [4] are not reported here, including the extension to `uniqueItems` for witness generations. The presentation of several steps (especially for witness generation) is driven/based by/on examples.

Also, we would like to stress that results presented in this paper takes part of research activities [4] that are still in progress. So our main aim here is to present existing results, mainly at the definition and formalisation level of algorithms.

3 RELATED WORK

The first effort to formalize the semantics of JSON Schema as by Pezoa et al. in [8] whose goal was to lay the foundations of the JSON schema proposal by studying its expressive power and the complexity of the validation problem. Along the lines of this work, Bouhris et al. [6] characterized the expressivity of the JSON Schema language and investigated the complexity of the satisfiability problem which turns out to be *2EXPTIME* in the general case and *EXPSPACE* when disallowing *uniqueItems*. None of the above works study the problem of generating an instance of a JSON Schema. The only attempt to solve this problem was investigated by Earle et. al [5] in the context of testing REST API calls but the presented solution, which is based on translating JSON Schema definitions into an Erlang expression, is not formally defined and restricted to atomic values, objects and to some form of boolean expressions.

From the point of view of schema normalization, the closest work to ours is the one in [7] which studies schema inclusion for JSON Schema. To cope with the high expressivity of the JSON Schema language, a pre-requisite step is needed to rewrite the schemas into a *Disjunctive Normal Form* which has some similarities with the preparation phase of our work. However, compared to our work, the schema normalization in [7] lacks the ability of eliminating negation for all kinds constraints, does not deal with recursive definitions and is not able to decide schema satisfiability which is captured by the *inhabited( )* predicate whose specification is only informally discussed. This has been confirmed in practice by experimenting the tool developed in [7] for parsing real world schemas described in [4]: the tool raised an issue for 21,859 out of 23,480 input schemas. The dominating error is related to constructs not being supported, but many other errors due to the inability to parse recursive schemas or to navigate references are present.

4 CONCLUSION

JSON Schema is an evolving standard for the description of families of JSON documents, and is widely used in data-centric applications. Despite the recent interest in the research community related to this schema language, crucial problems like schema equivalence/inclusion and consistency have either been partially dealt with or not explored at all. In this work we present our approach in order to solve these problems, based on our algebraic specification of JSON Schema. We are currently finalizing a Java implementation of the presented algorithm, and studying optimisation techniques, by analysing a large repository of JSON Schemas allowing us for determining how often mechanisms that are critical for execution times are used. We are also investigating witness generation techniques able to generate several instances meant be used for testing queries and programs manipulating valid JSON data.

ACKNOWLEDGEMENTS

The research has been partially supported by the MIUR project PRIN 2017FTXR7S "IT-MaTTerS" (Methods and Tools for Trustworthy Smart Systems).

REFERENCES

[1] Internet engineering task force, 2020. Available at https://www.ietf.org.
[2] Json schema, 2020. Available at https://json-schema.org.
[3] Json schema test suite, 2020. https://github.com/json-schema-org/JSON-Schema-Test-Suite/blob/master/tests/draft6/additionalProperties.json.
[4] Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, and Stefanie Scherzinger. Not elimination and witness generation for json schema. 2020. Available at https://webia.lip6.fr/~baazizi/rs/js/dism/witnessgen.pdf.
[5] Clara Benac Earle, Lars-Åke Fredlund, Ángel Herranz, and Julio Mariño. Jsongen: a quickcheck based library for testing json web services. In *Proceedings of the Thirteenth ACM SIGPLAN workshop on Erlang*, pages 33–41, 2014.
[6] Pierre Bourhis, Juan L. Reutter, Fernando Suárez, and Domagoj Vrgoc. JSON: data model, query languages and schema specification. In Emanuel Sallinger, Jan Van den Bussche, and Floris Geerts, editors, *PODS*, pages 123–135. ACM, 2017.
[7] Andrew Habib, Avraham Shinnar, Martin Hirzel, and Michael Pradel. Type safety with json subschema, 2019.
[8] Felipe Pezoa, Juan L. Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. Foundations of json schema. In *WWW '16*, pages 263–273, 2016.
[9] A. Wright, H. Andrews, and B. Hutton. JSON Schema validation: A vocabulary for structural validation of json - draft-handrews-json-schema-validation-02. Technical report, Internet Engineering Task Force, sep 2019.

[1] Example available at [3].

EPIQUE: A Graph Data Model and Query Language for Exploring the Evolution of Science

Ke Li
LIP6, CNRS, Sorbonne Université
Paris, France
ke.li@lip6.fr

Hubert Naacke
LIP6, CNRS, Sorbonne Université
Paris, France
hubert.naacke@lip6.fr

Bernd Amann
LIP6, CNRS, Sorbonne Université
Paris, France
bernd.amann@lip6.fr

CCS CONCEPTS

• **Computing methodologies** → **Topic modeling**; • **Information systems** → *Temporal data*; Data mining.

KEYWORDS

Topic Modeling, LDA, Science Evolution, Big data

Introduction. There is an increasing demand for practical tools to explore the evolution of scientific research published in bibliographic archives such as the Web of Science (WoS), ISTEX, arXiv or PubMed. The study of science evolution can help *philosophers and historians* of science [**?**] to test their theories with data, *researchers* to position their work in its scientific context, *industry* to evaluate the potential for innovation and technological transfer, *librarians* to classify scientific documents, etc. Revealing meaningful evolution patterns from document archives has many other applications and can be extended to synthesize narratives from datasets across multiple domains, including news stories, research papers, legal cases and works of literature [**?**].

In the interdisciplinary ANR EPIQUE project[1], we adopt the cognitive view of scientific evolution which assumes that the evolution only depends on the textual document contents (title, abstract, main contents) [**?**]. Whereas this choice reduces the expressive power by excluding the *social view* taking account of co-authorship and citation graphs [**? ?**], it also decreases the "social" bias and detects more easily possible interactions between scientific ideas and contributions, independently of any particular scientific community. Graph-based topic evolution analysis builds on topic evolution networks [**?**] which track complex temporal evolution dynamics by periodical topic discovery and similarity-based topic alignment. Figure 1 shows a snippet of a topic evolution graph extracted from the arXiv[2] corpus. The graph covers the periods between 2000 and 2006 decomposed into three overlapping time periods (3 year periods with one year overlap). Each topic is represented by a rectangle containing the top-10 topic terms obtained by an NLP document pre-processing step. *Emerging* terms are shown in green, *decaying* term boxes are colored in red, *stable* terms which exist both, in ancestor topics and in descendant topics, are in blue and *specific* terms which appear only in the current topic are in white. The thickness of the alignment edges reflects the similarity of the

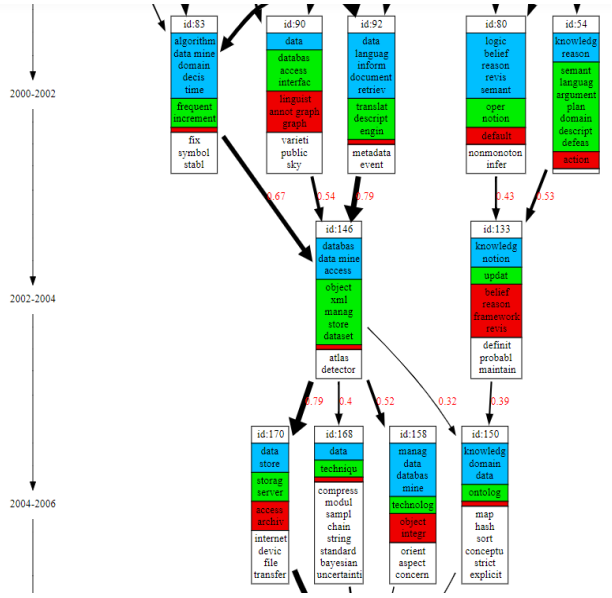


Figure 1: Pivot topics containing term "database" extracted from arXiv, green = emerging terms, blue = stable terms, red = decaying terms

connected topics. Several topics contain the term "database" and we can observe different evolution patterns. The topic evolution graph shows topics related to "data mining" (83), "data access interfaces" (90), "information retrieval" (92), "logics, semantics" (80) and "knowledge, reasoning" (54). The first three topics converge in 2002 – 2004 into a single topic on "object, xml, store, data mining" (146) which splits in the period of 2004 – 2006 into "storage servers" (170), "data mining and management" (158) and "knowledge and ontologies" (150).

Building and exploring topic evolution networks is still difficult and needs an important expertise in statistical text mining. A first challenge for domain experts is to correctly tune method specific hyper parameters with respect to a given dataset and an expected output. A second challenge concerns the visual exploration of large topic evolution networks. Whereas existing graph visualisation tools like Gephi [3] or Graphviz [4] can be used to generate high-quality visualisations, their use for exploring large graphs and identifying meaningful evolution patterns is difficult.

[1]This work was funded by French ANR-16-CE38-0002-01 project EPIQUE
[2]https://arxiv.org/

[3]https://gephi.org/
[4]https://www.graphviz.org/

Pivot Graph Model and Query Language. In this work we propose a data model for the visualisation and exploration of topic evolution networks representing the research progress in scientific document archives. Our model is independent of a particular topic extraction and alignment method and proposes a set of semantic and structural metrics for characterizing and filtering meaningful topic evolution patterns.

For identifying topic evolution patterns we decompose topic evolution graphs into subgraphs defined by a chosen topic t connected to other topics through alignment edges with some minimal similarity threshold β. Each couple (t, β) of some topic t and threshold β called a *pivot topic* and corresponds to a family of subgraphs $\mathcal{G}(t, \beta)$ called *pivot graphs*. We distinguish three particular pivot graphs denoted by (1) $\mathcal{G}^f(t, \beta)$, the maximal subgraph with all nodes that are reachable from t through paths with minimal edge weight β, (2) $\mathcal{G}^p(t, \beta)$, the maximal subgraph with all nodes that can reach t through paths with minimal edge weight β and their union (3) $\mathcal{G}^*(t, \beta) = \mathcal{G}^p(t, \beta) \cup \mathcal{G}^f(t, \beta)$.

The evolution of a topic t can then be characterized by the structure of its future $\mathcal{G}^f(t, \beta)$ and its past $\mathcal{G}^p(t, \beta)$ for different β-thresholds. The goal of our pivot graph model is to define a query language which allows users to filter topics according to some useful metrics concerning their evolution represented by their pivot graphs.

Our query language allows experts to filter pivot graphs according to some evolution pattern defined by the combination of graph evolution filters. For example query $Q1$ filters all pivot topics where the future has an average edge similarity (relative evolution degree) $Revol > 0.6$ and an average pivot topic similarity (pivot evolution degree) $Pevol > 0.5$, each future topic has two child topics in average (*Split*) and there exist future subtopics related to the pivot topic with a minimal distance of 5 periods (*Live*):

```
Q1: DB.Future.Revol(>=0.5).Pevol(>=0.6)
      .Split(>=2).Live(=5)
```

Observe that the user does not specify the β-threshold and the result contains for each topic t all its pivot topics (t, β) satisfying the filter.

Apart from these metric-based filters, our query language also allows users to define other multi-dimensional filtering criteria including topic labels and temporal conditions for the selection of pivot topics. For example, the following query *finds all topics with an emerging term "deep learning" where the past contains a path to a topic with the decaying term "big data"*:

```
Q2: DB.Emerge("deep learning")
      .Past.Path(Decay("big data"))
```

Finally, pivot topics and their associated metrics can be used for the structural and quantitative analysis of topic evolution graphs. For example Figure 2 shows the distribution of *future* pivot evolution graphs in arXiv with respect to their *split degree* and *convergence degree* . We can see that a low threshold $\beta = 0.2$ generates a large number of complex pivot topic graphs with high split and convergence degrees.

Implementation and Experimentation. The long version of this article includes a more detailed description of the underlying algorithms and other important aspects concerning quality issues like

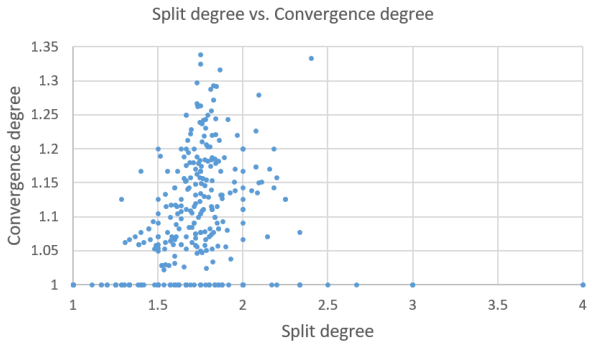


Figure 2: β = 0.2, #T = 50, #Pivot = 477, #Isolated = 23

topic diversity. The workflow also has been implemented on top of Apache Spark and we have have conducted several experiments on four real-world scientific archives covering 20 years of scientific publications including 1.15 million scientific articles extracted from arXiv and 1 million documents extracted from Wiley's Web Of Science.

Conclusion. In this article we propose a generic evolution network computation and visualization framework which combines a high-level data model with big data technology for extracting and exploring topic evolution networks. The graph model relies on the notion of *pivot topic graphs*, which describe the contents and the evolution dynamics of topics at different levels of detail. The model also includes a number of high-level semantic metrics which enable domain experts to specify meaningful topic evolution patterns (queries) for exploring large topic evolution networks. This framework has been completely implemented on top of Apache Spark using LDA and cosine similarity for topic extraction and topic alignment. The user can express complex evolution pattern queries to obtain the relevant pivot topic graphs. A first prototype [**?**] is used to extract complex evolution patterns for different scientific domains as part of the EPIQUE project and in collaboration with philosophers of science. As future work we intend to optimize the computation of pivot topic evolution graphs and exploit the LDA document-topic matrix for enriching the analysis.

REFERENCES

[] David Chavalarias and Jean-Philippe Philippe Cointet. 2013. Phylomemetic patterns in science evolution—the rise and fall of scientific fields. *PloS one* 8, 2 (2013), e54847.

[] Eugene Garfield. 1955. Citation Indexes for Science: A New Dimension in Documentation through Association of Ideas. *Science* 122, 3159 (July 1955), 108–111.

[] Thomas S. Kuhn, Otto Neurath, and Thomas Samuel Kuhn. 1994. *The Structure of scientific revolutions* (2nd ed., enlarged ed.). Number ed.-in-chief: Otto Neurath ; Vol. 2 No. 2 in International encyclopedia of unified science Foundations of the unity of science. Chicago Univ. Press, Chicago, Ill. OCLC: 258260085.

[] Ke Li, Hubert Naacke, and Bernd Amann. 2020. EPIQUE: Extracting Meaningful Science Evolution Patterns from Large Document Archives (Demonstration). In *Int'l Conf. on Extending Database Technology (EDBT)*. Copenhagen, Denmark.

[] Dafna Shahaf, Carlos Guestrin, Eric Horvitz, and Jure Leskovec. 2015. Information Cartography. *Commun. ACM* 58, 11 (2015), 62–73.

[] Xiaoling Sun, Jasleen Kaur, Staša Milojević, Alessandro Flammini, and Filippo Menczer. 2013. Social Dynamics of Science. *Scientific Reports* 3 (Jan. 2013), 1069. https://doi.org/10.1038/srep01069

Approche supervisée pour l'appariement d'entités dans le domaine Transport et Logistique

Yassine Guermazi
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
yassine.guermazi@lis-lab.fr

Sana Sellami
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
sana.sellami@lis-lab.fr

Omar Boucelma
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
omar.boucelma@lis-lab.fr

ABSTRACT

L'appariement d'entités (Entity Matching) est un problème crucial pour l'intégration de données. Il consiste à identifier des entités, ayant éventuellement des structures différentes, qui représentent une seule et même entité du monde réel. Dans ce travail, nous nous intéressons à l'appariement d'entités dans le domaine du Transport et Logistique. Aux difficultés usuelles qui caractérisent la problématique (typos, données manquantes ou redondantes, etc.), s'ajoutent deux autres problèmes : 1) L'appariement sémantique entre les adresses (les éléments d'adresse qui partagent le même contexte géographique peuvent être similaires) et 2) les spécificités « domaine » comme les abréviations et les acronymes dans les noms de sociétés ainsi que l'absence d'un format standard pour les adresses.

Nous distinguons deux catégories d'approches d'appariement d'entités. La première s'appuie, soit sur des techniques de similarité en combinaison avec des règles, soit sur des techniques d'apprentissage. Ces approches sont performantes pour l'appariement syntaxique sur des données peu bruitées mais ne tiennent pas compte d'un possible lien sémantique entre entités. La deuxième catégorie fait appel des techniques de représentations distribuées de mots (Word Embedding) en combinaison avec de l'apprentissage profond. Ces approches nécessitent un grand ensemble de données d'apprentissage et ne sont pas adaptées pour des jeux de données de faible taille.

Dans cet article, nous proposons un processus d'appariement en deux phases : 1) La standardisation des entités en vue de leur prétraitement et du parsing des adresses (données textuelles) et, 2) L'appariement par apprentissage supervisé sur des représentations vectorielles d'entités obtenues par des méthodes de Word Embedding. Les expérimentations ont été menées sur un jeu réel de données représentant des entités de Transport et Logistique en France. Les résultats d'évaluation, en comparaison avec d'autres méthodes ou un système comme Magellan, illustrent la performance de notre approche, notamment avec un jeu d'apprentissage de taille réduite.

Discovery of Link Keys in RDF Data Based on Pattern Structures: Preliminary Steps

Nacira Abbas
nacira.abbas@inria.fr
Université de Lorraine, CNRS, Inria, Loria
Nancy, France

Jérôme David
jerome.david@inria.fr
Université de Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG
Grenoble, France

Amedeo Napoli
amedeo.napoli@loria.fr
Université de Lorraine, CNRS, Inria, Loria
Nancy, France

ABSTRACT

In this paper, we are interested in discovering link keys among two different RDF datasets. We introduce a specific and original pattern structure where link keys can be discovered in one pass while specifying the pair of classes associated with each link key.

CCS CONCEPTS

• **Computing methodologies** ↑ *Knowledge representation and reasoning*;

KEYWORDS

link key discovery, pattern structures, linked data

Data interlinking is the task of finding identity links across RDF datasets. Given two RDF datasets D_1 and D_2, we aim to discover identity links between them. An identity link is a statement of the form $\langle a, \texttt{owl:sameAs}, b\rangle$ expressing that the resource a from D_1 and the resource b from D_2 represent the same real world entity. For short, we write $\langle a,b\rangle$ and we call this pair a *link*. A link key is used to generate such links. It takes the form of two sets of pairs of properties Eq and In associated with a pair of classes. The pairs of properties express sufficient conditions for two resources, from the associated pair of classes, to be the same. An example of a link key is $k = (\overbrace{\{\langle \texttt{designation}, \texttt{title}\rangle\}}^{\text{Eq}}, \overbrace{\{\langle \texttt{designation}, \texttt{title}\rangle, \langle \texttt{creator}, \texttt{author}\rangle\}}^{\text{In}}, \langle \texttt{Book}, \texttt{Novel}\rangle)$. It states that whenever an instance a of the class `Book` has the same values for the property `designation` as an instance b of the class `Novel` for the property `title`, and that a and b share at least one value for the properties `creator` and `author`, then k generates the link $\langle a,b\rangle$. As the properties in RDF are not necessarily functional, the property values are compared in different ways: the resources share all of their values for the pairs of properties in Eq and they share at least one of them for the pairs of properties in In.

Algorithms [2, 3] have been proposed to discover link key candidates from datasets. These candidates are then evaluated and the best ones are used to find identity links. A link key candidate is defined as a link key that generates at least a link and it is maximal[1] on the links that it generates. The quality of a link key candidate k is evaluated using *coverage* and *discriminability* [2]: the *coverage* measures how general k is and the *discriminability* measures the capability of k to discriminate between resources. In order to take into account these two measures, an harmonic mean might be used.

The existing algorithms for link keys discovery take as input two datasets ignoring the classes to which the resources belong to. As a result link key candidates are returned without specifying their pairs of classes. These candidates are then evaluated considering the whole datasets. However, this evaluation is not accurate since a link key candidate may be relevant for a pair of classes and not relevant for another pair. In this paper[2], we propose a method based on pattern structures [4], a generalization of Formal Concept Analysis (FCA), that overcomes these limits. This method allows to find link key candidates while specifying their associated pairs of classes. The term 'class' here refers to an atomic class or to a class expression defined here as a disjunction $\sqcup_{DL}$ of conjunction $\sqcap_{DL}$ of classes as in description logics. The use of pattern structures is motivated by the fact that the definition of a link key candidate matches the definition of a closed set and the aim FCA is to discover such sets. We define the pattern structure for link key candidates discovery, called the $LK-$*pattern structure* as follows. Given two datasets D_1 and D_2 where the sets $S(D_1), S(D_2)$ denote respectively the set of subjects in D_1, D_2. The $LK-$*pattern structure* for D_1 and D_2 is the triple $(S(D_1) \times S(D_2), (E, \sqcap), \delta)$ where the set of objects $S(D_1) \times S(D_2)$ is the set of pairs of subjects over D_1 and D_2. E is the set of potential object descriptions. A description is a link key k over D_1 and D_2. $(E, \sqcap)$ is a semilattice where the meet $\sqcap$ of two descriptions k_1 and k_2 is a link key k such as its set Eq (resp.In) is the intersection of the sets Eq (resp.In) of k_1 and k_2 and its associated pair of classes is the disjunction $(\sqcup_{DL})$ of the pairs of classes of k_1 and k_2. The descriptions are partially ordered by $\sqsubseteq$ defined w.r.t. the similarity operator $\sqcap$. If $k_1 \sqcap k_2 = k_1 \Leftrightarrow k_1 \sqsubseteq k_2$. The mapping $\delta : S(D_1) \times S(D_2) \to E$ associates each pair of subjects $\langle s_1, s_2\rangle \in S(D_1) \times S(D_2)$ to its description k which is the maximal link key generating the link $\langle s_1, s_2\rangle$. Figure 1 represents an example of a pattern concept lattice generated from an $LK-$pattern structure.

[1] The definition of maximality in link keys is given in [3].
[2] We originally published this work in the CLA conference proceedings [1].

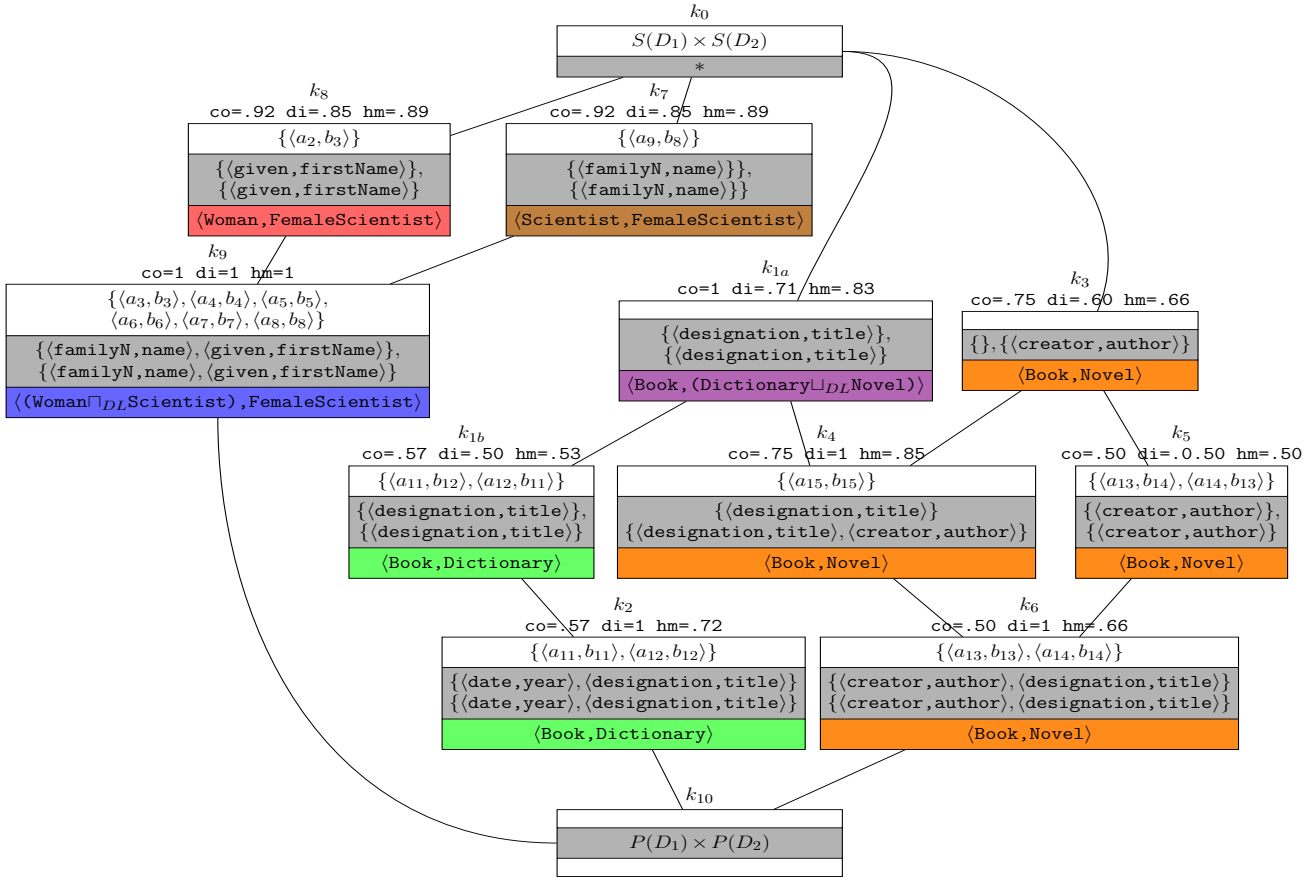


Figure 1: An example of a pattern concept lattice generated from an $LK-$pattern structure. The harmonic mean of the coverage co and the discriminability di is denoted by hm.

Each node here represents a pattern concept which is a pair $(L(k), k)$ such as $L(k)$ and k are respectively the extent and the intent of this pattern concept (closed sets). The intents are link key candidates and the extents are sets of links genetated by their intents. For example the link key k_{1b} = ({⟨designation,title⟩},{⟨designation,title⟩}⟨Book,Dictionary⟩) generates the set of links $L(k_{1b}) = \{\langle a_{11}, b_{12}\rangle, \langle a_{12}, b_{11}\rangle, \langle a_{11}, b_{11}\rangle, \langle a_{12}, b_{12}\rangle\}$. We can see in the lattice that k_{1b} is associated with the pair of classes ⟨Book,Dictionary⟩. Whereas this was not possible using other algorithms [2, 3]. As a matter of fact, specifying the pairs of classes associated with a link key candidate is a critical task to properly evaluate this candidate. For example, the link key candidate k_4 has shown a low harmonic mean when evaluated on the whole datasets, consequently, k_4 will not be returned as a relevant candidate despite the fact that it generates all the links between the classes Book and Novel while no other candidate is able to generate those links. By contrast, in Figure 1, k_4, shows a good harmonic mean because it is evaluated on the "right pair" of classes ⟨Book,Novel⟩. Hence, we can appreciate the importance of introducing the notion of $LK-$pattern structure and the discovery of link key candidates associated with pairs of classes.

ACKNOWLEDGMENTS

This work has been supported by the ANR project Elker (ANR-17-CE23-0007-01) and the BnF in the context of the agreement between Inria and Ministère de la culture.

REFERENCES

[1] Nacira Abbas, Jérôme David, and Amedeo Napoli. 2020. Discovery of Link Keys in RDF Data Based on Pattern Structures: Preliminary Steps. In *Proceedings of CLA*. 235–246.
[2] Manuel Atencia, Jérôme David, and Jérôme Euzenat. 2014. Data interlinking through robust linkkey extraction. In *ECAI*. 15–20.
[3] Manuel Atencia, Jérôme David, Jérôme Euzenat, Amedeo Napoli, and Jérémy Vizzini. 2020. Link key candidate extraction with relational concept analysis. *DAM*. 273 (2020), 2–20.
[4] Bernhard Ganter and Sergei O Kuznetsov. 2001. Pattern structures and their projections. In *ICCS*. Springer, 129–142.

5 Résumés des articles courts

Towards application-specific query processing systems

Dimitrios Vasilas
Scality
Sorbonne Université - LIP6 & Inria
dimitrios.vasilas@lip6.fr

Marc Shapiro
Sorbonne Université - LIP6 & Inria
marc.shapiro@acm.org

Bradley King
Scality
brad.king@scality.com

Sara S. Hamouda
Sorbonne Université - LIP6 & Inria
sara.hamouda@inria.fr

ABSTRACT

Database systems use query processing sub-systems for enabling efficient query-based data retrieval. An essential aspect of designing any query-intensive application is tuning the query system to fit the application's requirements and workload characteristics. However, the configuration parameters provided by traditional database systems do not cover the design decisions and trade-offs that arise from the geo-distribution of users and data. In this paper, we present a vision towards a new type of query system architecture that addresses this challenge by enabling query systems to be designed and deployed in a per use case basis. We propose a distributed abstraction called Query Processing Unit that encapsulates primitive query processing tasks, and show how it can be used as a building block for assembling query systems. Using this approach, application architects can construct query systems specialized to their use cases, by controlling the query system's architecture and the placement of its state. We demonstrate the expressiveness of this approach by applying it to the design of a query system that can flexibly place its state in the data center or at the edge, and show that state placement decisions affect the trade-off between query response time and query result freshness.

Experimental study of regret minimization sets and multidimensional Skylines

Karim Alami
Univ. Bordeaux, CNRS, LaBRI, UMR 5800
Talence, France
karim.alami@u-bordeaux.fr

Sofian Maabout
Univ. Bordeaux, CNRS, LaBRI, UMR 5800
Talence, France
sofian.maabout@u-bordeaux.fr

ABSTRACT

Skyline and Top-K operators are both multi-criteria preference queries. The advantage of one is a limitation of the other: Top-k requires a scoring function while Skyline does not, and Top-k output size is exactly K objects while Skyline's output can be the whole dataset. To cope with this state of affairs, regret minimization sets (RMS) whose output is bounded by K and where there is no need to provide a scoring function has been proposed in the literature. However, the computation of RMS on top of the whole dataset is time-consuming. Hence some previous works proposed the Skyline instead of the whole data as input. This optimization while it guarantees the result correctness, it becomes of no interest when the skyline itself is large, e.g. with anti-correlated and/or high dimensionality. In this paper we investigate the speedup provided by other Skyline related candidate, e.g., Top-K frequent Skylines thanks to NSC, an index structure we have proposed in earlier works. Our empirical results show that these candidate sets provide interesting execution time/result quality tradeoff as a solution for computing RMS.

KEYWORDS

Query optimization, Regret Minimization sets, Multidimensional Skylines, Top-K

Leveraging Change Point Detection for Activity Transition Mining in the Context of Environmental Crowdsensing

Hafsa El Hafyani
DAVID Lab
UVSQ - Université Paris-Saclay
hafsa.el-hafyani@uvsq.fr

Karine Zeitouni
DAVID Lab
UVSQ - Université Paris-Saclay
karine.zeitouni@uvsq.fr

Yehia Taher
DAVID Lab
UVSQ - Université Paris-Saclay
yehia.taher@uvsq.fr

Mohammad Abboud
Lebanese University
mohammad.abboud.2496@gmail.com

ABSTRACT

The change point detection is a critical problem in time series analysis. Detecting these transitions is gainful to human activities recognition. In this paper, we leverage this method to discover the transition between activities based on data originated from different sensors. We design and evaluate a change point detection process for the environmental crowd sensing data. We detect transitions and integrate the change point detection with multi-dimensional time series to enhance the time series segmentation into separate activities. Experiments on real-world environmental crowd sensing data suggest that combining different dimensions lead to higher performance for the change points detection.

KEYWORDS

Activity Recognition, Change point detection, Segmentation, Data Mining, Mobile Crowd Sensing

1 INTRODUCTION

With the rapid advances of Internet of Things (IoT), along with the widespread use of GPS, and other built-in and external environmental sensors, several applications have emerged to collect geodated data series. One of such applications is the new paradigm of Mobile Crowd Sensing (MCS), which empowers volunteers to contribute data acquired by their personal sensor-enhanced mobile devices [2]. Polluscope[1], a french project deployed in Île-de-France (i.e., Paris region), is a typical use case study based on MCS. It aims at getting insight constantly on individual exposure to pollution everywhere (indoor and outdoor), while enriching the traditional monitoring system with the collected data by the crowd. The recruited participants, on a voluntary basis, collect air quality measurements such as Particulate Matters, NO2, Black Carbon, Temperature and Humidity. Each participant is equipped with a sensor kit and a mobile device which allows for the transmission of collected measurements together with their GPS coordinates as geo-dated data series, and activity annotation through a custom mobile application. This paradigm will allow participants to have personalized insights about their exposures to pollution. It measures indoor and outdoor environments (e.g., Home, Work, Transportation, Streets, Park, etc), and enables participants to gain insights at a higher resolution along their trajectories, thereby, allowing to capture local variability and peaks of pollution, depending on participants' whereabouts.

It is worth mentioning that the ambient air observations strongly depend on the context. More than that, it could be a proxy for an indoor versus outdoor environment. For this reason, there is a great interest of making the analysis context-aware. For this to be done, we need to identify the context automatically from the collected raw data.

Since the context changes with the participants' activity and whereabouts, this also means that we need to detect the changes and segment the geo-data series into non overlapping segments according to participants micro-environments (i.e. Home, Work, Transportation, Streets, Park, etc.). Segmented data is a prerequisite for activity recognition mining task, which assigns to each segment a labeled with a single activity.

This extended abstract introduces our proposal in [4] of a multi-dimensional time series segmentation to discover activities and events boundaries in the context of mobile crowd sensing. The main contribution is precisely the combination of different dimensions in the change point detection, when not all dimensions may cause or contribute evenly in discovering the change in participant's activities or events. Our approach combines data pre-preparation, change point detection on individual dimensions, and a post-processing phase to fuse the detection from multiple dimensions. This last phase is based on a supervised learning approach. We implement and test our framework in a real-application setting. Please refer to the original paper [4] form more details.

2 CHANGE POINT DETECTION MODEL

In this section, we introduce briefly our change point detection approach based on CUSUM algorithm for multi-dimensional time series in environmental crowd sensing. We refer the interesting readers to our full paper [4] where both the theory and the implementation aspects of our approach are discussed formally.

The implementation of our proposed process includes four parts: data collection and preparation, change point detection, post-processing and ensemble method learning.

[1] http://polluscope.uvsq.fr

2.1 Data Collection and Preparation

A sequence of these data contains timestamp, sensor kit ID, Latitude, Longitude, ambient air data (in our case, Temperature, Humidity, PM2.5, PM10, PM1.0, NO2 and Black Carbon), activity (Car) and events (such as "open a window").

Participants geo-locations are collected as GPS logs. We drive the velocity time series from GPS coordinates.

2.2 Change Point Detection

The change point detection problem is the process of detecting abrupt changes in time series data. The change points are detected when the probability distribution of time series changes abruptly between two consecutive intervals.

The overall question is: how to combine all these different aspects of the data (geo-location, sensors, partially annotated activities and events) to segment and discover the context of the user, and to discriminate the observations in different micro-environment ? This is called a holistic approach of activity recognition [3].

The segmentation phase consists mainly in splitting spatio-temporal data into coherent segments. Each segment represents a micro-environment. One way to do this segmentation is to detect the changes either in the ambient time series, or in the geo-location. The former corresponds to the problem of change point detection (CPD) in time series. Many solutions exist in the literature when it comes to mono-variate time series [1]. As for the the GPS data, it is related to the so-called stop & move detection in trajectories [6–9]. In this paper, we use the change point detection in time series for both problems, simply by adding the velocity dimension, which is easily derived from geo-location data.

2.3 Post-Processing

In multi-dimensional time series, some dimensions may contribute more in the explanation of the change, while others may be considered as irrelevant. Plus, we have made this observation that participant's context is very highly correlated with ambient air temperature and humidity more than, for example, speed.

It is very unlikely for the detected change point from all the dimensions to be at exactly the same timestamp.Post-processing the CUSUM [5] algorithm results will improve the change point detection accuracy by merging the detected change points into one change point if a certain condition is verified.

2.4 Ensemble Method Learning

One of the contribution of this work is the combination of multi-dimensional sensory time series data and geo-located data (i.e. GPS data) to detect the changes boundaries of participants micro-environments when some dimensions may be considered irrelevant to the change detection or not all dimensions cause the change. In order to enhance the accuracy of the change point detection, many ensemble methods have been proposed to further enhance the algorithms accuracy by combining learners rather than trying to find the best single learner [10] .

In this work, we propose a model that integrates the CUSUM change point detection algorithm with multi-dimensional time series to achieve a strong combination abilities. The model works as follows: (1) the change point detection algorithm is applied on each time series dimension separately; (2) each dimension generates a set of detected change points, with a certain accuracy to the ground truth; (3) the weights of every dimension are then learned from the gold set data annotated by activities and events of participants. The model used in this experiment include: *AdaBoost* with Decision Tree, *Decision Tree* (DT), *SVM* and *Logistic Regression*. The proposed model allows to understand which dimension is affected by the changes in participants micro-environments and pollution related events.

3 CONCLUSION

Change point detection segmentation can provide insights about human behaviour's transition. Participants' whereabouts can be learned after segmenting the collected multi-dimensional time series, and discover insights about individual exposure to pollution.

In this abstract, we have summarized a change point detection approach based on the *Cumulative Sum* algorithm to discover transition points in multi-dimensional time series using real world data collected in the context of environmental crowd sensing.

The experiment conducted in multi-dimensional time series, where not all dimensions may cause the change, has shown that our approach outperforms the traditional CUSUM algorithm, using *AdaBoost* as a combining learner algorithm [4].

ACKNOWLEDGMENTS

This work is supported by the grant ANR-15-CE22-0018 Polluscope of the French National Research Agency (ANR). This work has also received a co-financing from DIM Qi^2 with the support of the Ile-de-France region.

REFERENCES

[1] Samaneh Aminikhanghahi and Diane J. Cook. 2016. A survey of methods for time series change point detection. *Knowledge and Information Systems* 51 (2016), 339–367.

[2] Mariem Brahem, Hafsa E.L. Hafyani, Souheir Mehanna, Karine Zeitouni, Laurent Yeh, Yehia Taher, Zoubida Kedad, Ahmad Ktaish, Mohamed Chachoua, and Cyril Ray. 2021. Data perspective on environmental mobile crowd sensing. In *Intelligent Environmental Data Monitoring for Pollution Management*. Academic Press, 269 – 288.

[3] Hafsa El Hafyani. 2020. Big Data Series Analytics in the Context of Environmental Crowd Sensing. *The IEEE International Conference on Mobile Data Management* (2020).

[4] Hafsa El Hafyani, Karine Zeitouni, Yehia Taher, and Mohammad Abboud. 2020. Leveraging Change Point Detection for Activity Transition Mining in the Context of Environmental Crowdsensing. *The 9th SIGKDD International Workshop on Urban Computing* (2020).

[5] E. S. Page. 1954. Continuous Inspection Schemes. *Biometrika* 41 (1954), 100–115.

[6] Benoît Thierry, Basile Chaix, and Yan Kestens. 2013. Detecting activity locations from raw GPS data: a novel kernel-based algorithm. In *International Journal of Health Geographics*.

[7] Apichon Witayangkurn, Teerayut Horanont, Natsumi Ono, Yoshihide Sekimoto, and Ryosuke Shibasaki. 2013. Trip reconstruction and transportation mode extraction on low data rate GPS data from mobile phone. (2013), 1–19.

[8] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei-Ying Ma. 2010. Understanding Transportation Modes Based on GPS Data for Web Applications. *TWEB* 4 (01 2010).

[9] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. 2008. Understanding mobility based on GPS data. In *UbiComp*.

[10] Zhi-Hua Zhou. 2012. Ensemble Methods: Foundations and Algorithms. CRC press.

6 Résumés des articles de démonstration

Task-Tuning in Privacy-Preserving Crowdsourcing Platforms

Joris Duguépéroux
Univ Rennes, CNRS, IRISA
Rennes, France
joris.dugueperoux@irisa.fr

Antonin Voyez
Univ Rennes, CNRS, IRISA
Rennes, France
antonin.voyez@irisa.fr

Tristan Allard
Univ Rennes, CNRS, IRISA
Rennes, France
tristan.allard@irisa.fr

Abstract

Specialized worker profiles of crowdsourcing platforms may contain a large amount of identifying and possibly sensitive personal information (e.g., personal preferences, skills, available slots, available devices) raising strong privacy concerns. This led to the design of privacy-preserving crowdsourcing platforms, that aim at enabling efficient crowdsourcing processes while providing strong privacy guarantees even when the platform is not fully trusted. We propose a demonstration of the PKD algorithm, a privacy-preserving space partitioning algorithm dedicated to enabling secondary usages of worker profiles within privacy-preserving crowdsourcing platforms by combining differentially private perturbation with additively-homomorphic encryption. The demonstration scenario showcases the PKD algorithm by illustrating its use for enabling requesters tune their tasks according to the actual distribution of worker profiles while providing sound privacy guarantees.

Obi-Wan: Ontology-Based RDF Integration of Heterogeneous Data

Maxime Buron
Inria and Institut Polytechnique de Paris
France
maxime.buron@inria.fr

François Goasdoué
Univ. Rennes, CNRS, IRISA
France
fg@irisa.fr

Ioana Manolescu
Inria and Institut Polytechnique de Paris
France
ioana.manolescu@inria.fr

Marie-Laure Mugnier
Univ. Montpellier, LIRMM, Inria
France
mugnier@lirmm.fr

ABSTRACT

The proliferation of digital data sources in many domains brings a new urgency to the need for tools which allow to flexibly query heterogeneous data (relational, JSON, key-values, graphs etc.) Traditional data integration systems fall into two classes: *data warehousing*, where all data source content is materialized in a single repository, and *mediation*, where data remains in their original stores and all data can be queried through a *mediator*.

We propose to demonstrate Obi-Wan, a novel mediator following the Ontology-Based Data access (OBDA) paradigm. Obi-Wan integrates data sources of many data models under an interface based on RDF graphs and ontologies (classes, properties, and relations between them). The novelty of Obi-Wan is to combine maximum integration power (GLAV mappings, see below) with the highest query answering power supported by an RDF mediator: RDF queries not only over the data but also over the integration ontologies. This makes it more flexible and powerful than comparable systems.

Acknowledgements: This work is supported by the Inria Project Lab iCoda and by ANR-18-CE23-0003.

How to Implement NoSQL Schemas with ModelDrivenGuide?

Jihane Mali
jihane.mali@usmba.ac.ma
Université Sidi Mohamed Ben Abdellah
Fès, Morocco

Faten Atigui
faten.atigui@cnam.fr
CEDRIC, Conservatoire National des Arts et Métiers (CNAM)
Paris, France

Ahmed Azough
ahmed.azough@usmba.ac.ma
Université Sidi Mohamed Ben Abdellah
Fès, Morocco

Nicolas Travers
nicolas.travers@devinci.fr
Léonard de Vinci Pôle Universitaire, Research Center
Paris La Défense, France

ABSTRACT

With the evolution of data in terms of volume, variety and velocity, designing and developing an Information Systems (IS) requires studying the best solutions to store and manipulate data while respecting the user's requirements. In this demonstration, we show how to implement an IS using ModelDrivenGuide, which is a semi-automated approach based on transformation rules starting from a conceptual model, then going from one logical model to an other by refinement to finally the chosen physical model.

KEYWORDS

NoSQL, MDA, Meta-Model, Model Transformation, Model Refinement, Ecore, QVT

1 INTRODUCTION

For decades, the storage and the exploitation of data has mainly relied on relational databases. With the advent of Big Data, the volume of data has exploded, the heterogeneity has increased tenfold, causing problems of transformation from traditional databases to new storage on the Cloud, whether in terms of storage management, data query, cost or performance. To deal with these problems, *NoSQL* data management systems have appeared since 2009.

Several works have focused on storage and modeling problems of data using NoSQL systems. Most of the studies have proposed either (i) a comparative study between RDB (Relational DataBases) and NoSQL DB (DataBases) and/or how to transform relational data into dedicated NoSQL system [6, 9, 11] or (ii) how to transform a conceptual schema into a specific NoSQL DB [1, 3–5] (iii) while very few studies have proposed criteria for the choice of physical models and implementation platforms [8, 10].

Our ModelDrivenGuide [2, 7] approach offers logical modeling suitable for models refinement in order to generate all types of optimized physical models by relying on a common 5Families meta-model (4 NoSQL families & the Relational model). Based on transformation rules, it provides a functional process that integrates the use case to generate the different SQL and/or NoSQL solution(s) adapted to business requirements.

2 MODELDRIVENGUIDE: FROM CONCEPTUAL MODEL TO PHYSICAL MODELS

We suggest a model driven approach that offers steps to generate a logical model for each family. Our approach focuses on model transformation, starting from conceptual to logical then to physical models, and on model refinement to go from one logical model to another. The peculiarity of this approach is to allow the optimization of the data model directly during the transformation process instead of the last step. This helps to make a choice of implementation according to the context of use. In order to formalize the process, we adopted a Model-Driven Architecture (MDA[1]).

Our ModelDrivenGuide approach (Figure 1) starts from a UML class diagram and considers two Platform Independent levels corresponding to the conceptual (PIM1) and logical (PIM2) levels, as well as a Platform Specific level related to the different target platforms.

The **PIM1** (*Platform Independent Model*) of the first level is an UML class diagram that serves as a basis for modeling the user requirements and the context of use.

The **PIM2** is the second level independent model, common to the five families of models. It allows to carry out *refinement* rules by generating recursively all possible denormalized models by relying on merge and split rules. We mention that a heuristic is applied to guide the generation of data models in order to avoid cycles and useless models. It is mainly based on the use case. This heuristic is not detailed in this demonstration.

The **PSMx** (*Platform Specific Model*) are obtained by the transformation of PIM2 models into the compatible target data family (e.g. nesting for DO, rows for CO, edges for GO, etc.).

This demonstration focuses especially on the PIM2 5Families meta-model used to generate all possible data models, on transformation, and on refinement rules.

2.1 Experimental Environment.

Since our approach is based on (MDA[2]), we need an infrastructure suitable for meta-modeling, modeling and models transformations. We developed our approach using a model transformation environment called *Eclipse Modeling Framework* (EMF[3]). It is a set of Eclipse plug-ins which can be used to design a data model and to generate code or other output based on this model. EMF respects

[1]MDA: https://www.omg.org/mda/index.htm
[2]MDA: https://www.omg.org/mda/index.htm
[3]EMF: https://www.eclipse.org/modeling/emf/

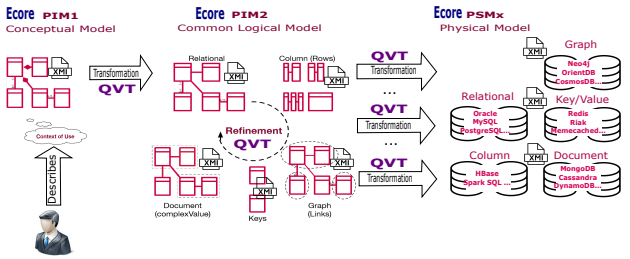


Figure 1: `ModelDrivenGuide` Approach

the known distinction between a meta-model and a model. A meta-model describes the structure of a model. A model is a concrete instance of this meta-model. To implement our approach, we have used the following tools proposed by EMF: 1) *Ecore* used for the implementation of all the `PIM1`, `PIM2`, and `PSMx` meta-models. Inspired by the object-oriented approach, the Ecore language is based on the notion of package (EPackage), class (EClass), attribute (EAttribute), reference link (EReference), data type (EDataType), and enumeration (EEnum); 2) *XMI*[4] a format in which instances of meta-models are created; and 3) *QVTO* (QVT Operational) a Model-To-Model (M2M) transformation tool that implements the QVT language. It is used to formalize both exogenous transformation rules (from conceptual to logical model, and from logical to physical model) and also refinement (endogenous transformation) rules.

3 DEMONSTRATION

Before starting the transformation process, it is necessary to define the three meta-models that we have in our `ModelDrivenGuide` approach since they are going to be needed in any further steps. We defined the conceptual (class diagram), logical (`5Families`) and physical (MongoDB) meta-models using `Ecore`. We also formalized the transformation/refinement rules in QVT.

3.1 TPC-C Benchmark Scenario

For this scenario, we have used the TPC-C[5] benchmark as an entry. It gives a full use case mixing at the same time transactions, joins and aggregations. The TPC-C benchmark simulates the behavior of a logistic DB on user orders with transaction-oriented stock management (OLTP). We will focus on the six classes (*Warehouse*, *District*, *Customer*, *Order* and *OrderLine*, *Item*) in the PIM1.

(1) Instantiate the PIM1 meta-model using TPC-C benchmark's class diagram,
(2) Typically, transform the PIM1 into the PIM2 as a normalized relational model conform to our `5Families` common meta-model,
(3) Apply applied semi-automatic refinement rules recursively,
(4) Transform the chosen model obtained from the refinement into a `MongoDB` database. The choice was led by the fact that `MongoDB` is one of the rare NoSQL solutions integrating ACID transactions in *sharding* (version 4.2) required by the TPC-C benchmark. However, our approach is extensible by defining a new PSM for each target database type (without ACID properties in that case). We can also visualize the JSON schema of the final output.

3.2 User Model Scenario

This scenario illustrates the whole process by integrating user's own data model. The user has to follow the undermentioned steps while using the `ModelDrivenGuide` Approach:

(1) Create its own instance of the source meta-model in the XMI format, in our case the source is the conceptual class diagram meta-model,
(2) Typically, transform the PIM1 into the PIM2 as a normalized relational model,
(3) Apply the heuristic to generate a tree of denormalized data models (using recursively refinement rules),
(4) Choose one or more generated PIM2, and apply its transformation into the target PSM.

4 CONCLUSION

Our `ModelDrivenGuide` approach is a MDA-based approach that aims to improve model transformation. `ModelDrivenGuide` is a global approach that generates optimized models based on the common `5Families` meta-model favoring the application of refinement rules to produce potential target models. This mix between data modeling and optimization rises to an approach that aims to find the efficient target model among the 5 families of data.

For future works, we seek to define a generic cost model allowing to compare the produced solutions and eventually suggest top-k logical models. This cost model will integrate different dimensions (storage, bandwidth, CPU/energy impact, etc.) and will help to make a decision among all produced data models.

[4] *XML Metadata Interchange*: https://www.omg.org/spec/XMI/About-XMI/
[5] TPC-C: http://www.tpc.org/TPCC/default5.asp

REFERENCES

[1] Fatma Abdelhedi, Amal Ait Brahim, Faten Atigui, and Gilles Zurfluh. 2017. MDA-based Approach for NoSQL Databases Modelling. In *International Conference on Big Data Analytics and Knowledge Discovery*. Springer, 88–102.
[2] Faten Atigui, Asma Mokrani, and Nicolas Travers. 2020. DataGuide : une approche pour l'implantation de schémas NoSQL. *Extraction et de Gestion des Connaissances (EGC'20)* RNTI-E-36 (2020), 407–408.
[3] G. Daniel, A. Gómez, and J. Cabot. 2019. UMLto[No]SQL: Mapping Conceptual Schemas to Heterogeneous Datastores. In *2019 13th International Conference on Research Challenges in Information Science (RCIS)*. 1–13.
[4] Gwendal Daniel, Gerson Sunyé, and Jordi Cabot. 2016. UMLtoGraphDB: mapping conceptual schemas to graph databases. In *International Conference on Conceptual Modeling*. Springer, 430–444.
[5] Shady Hamouda and Zurinahni Zainol. 2017. Document-oriented data schema for relational database migration to NoSQL. In *2017 International conference on big data innovations and applications (innovate-data)*. IEEE, 43–50.
[6] Chongxin Li. 2010. Transforming relational database into HBase: A case study. In *2010 IEEE international conference on software engineering and service sciences*. IEEE, 683–687.
[7] Jihane Mali, Ahmed Azgouh, Faten Atigui, and Nicolas Travers. 2020. DataGuide: An Approach for Implementing NoSQL Schemas. In *DEXA'20*. Bratislava, Slovakia, 1–10.
[8] AB Raut. 2017. NoSQL database and its comparison with RDBMS. *International Journal of Computational Intelligence Research* 13, 7 (2017), 1645–1651.
[9] Leonardo Rocha, Fernando Vale, Elder Cirilo, Dárlinton Barbosa, and Fernando Mourão. 2015. A framework for migrating relational datasets to NoSQL. *Procedia Computer Science* 51 (2015), 2593–2602.
[10] Clarence JM Tauro, Shreeharsha Aravindh, and AB Shreeharsha. 2012. Comparative study of the new generation, agile, scalable, high performance NOSQL databases. *International Journal of Computer Applications* 48, 20 (2012), 1–4.
[11] Tamás Vajk, Péter Fehér, Krisztián Fekete, and Hassan Charaf. 2013. Denormalizing data into schema-free databases. In *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*. IEEE, 747–752.

Scrutinizer: A System for Checking Statistical Claims

Georgios Karagiannis† Mohammed Saeed‡ Paolo Papotti‡ Immanuel Trummer†

†Cornell University, USA ‡EURECOM, France

1 INTRODUCTION

Data is often disseminated in the form of textual reports summarizing statistics. For authors of such documents, it is time-consuming and tedious to ensure the correctness of every claim. Nevertheless, erroneous claims about data are not acceptable in many scenarios as each mistake can have dire consequences, such as retractions or legal implications. We demonstrate Scrutinizer, a system that helps teams of fact checkers to verify consistency of text w.r.t. data.

The Scrutinizer system is inspired by a collaboration with the International Energy Agency (IEA). This NGO regularly publishes scientific reports encompassing hundreds of pages, requiring verification efforts by internal teams of experts. Using real reports and data, we will demonstrate how Scrutinizer reduces verification overheads in this scenario. More recently, we have published an online version of our system that verifies single statistical claims about the spread and effects of the coronavirus (https://coronacheck.eurecom.fr). This version attracts hundreds of daily users and we will use it for our demonstration as well.

Example 1.1. Consider a Tweet containing a textual claim stating "U.S. death rate is 1.3% in March 18" (https://twitter.com/liz_wheeler/status/1240789238628540416). There are multiple online sources of official data for the virus outbreak that can be used to demonstrate that it is incorrect. However, a content moderator would have to find the relevant dataset and manually write a query over such data to collect the relevant information. In the example:

```
SELECT b.March/a.March
FROM totalCases a, totalDeaths b
WHERE a.Country = 'USA', b.Country = 'USA'
```

Finally, the expert compares the output of the query with the claim and can eventually flag the content as incorrect.

Gathering data for the claim at hand and composing the right query for the validation takes expertise over the domain and data skills, typically taking minutes for a single claim. To reduce this time, given a document with statistical claims and related datasets, our system helps users to translate claims into corresponding SQL queries, to verify, and to potentially correct them.

We present Scrutinizer [3, 4], a system that analyzes claims via three primary methods: machine learning (ML) and natural language processing (NLP) for analyzing claim text, feedback from human domain experts for validating candidate queries, and query generation based on a large library of functions. The interpretable SQL queries are finally exposed to users so that they can either validate or flag as false the claim at hand.

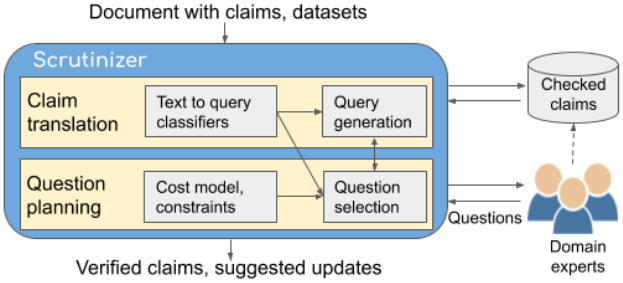


Figure 1: Architecture of Scrutinizer.

While prior verification systems [1, 2] assume that a single user verifies a short document based on a single data source, Scrutinizer is targeted at the verification of many, complex claims by teams of fact checkers. For instance, it learns to recognize new types of claims and queries as more claims from a given domain are verified. It supports claim queries connecting multiple data sources or containing complex, arithmetic expressions. Also, it uses a cost-based optimizer to plan the verification of large documents to minimize manual overheads. We describe next how the system can enable these features.

2 SYSTEM OVERVIEW

Figure 1 shows an overview of Scrutinizer. The input consists of a text document, containing one or more claims, and a set of relations. If a database of previously checked claims is available, our system uses it for bootstrapping. If no such database is available, we use active learning to steer the users in its creation. The output is a verification report, mapping verified claims to queries while pointing out mistakes and potential updates to the text, as shown in the example in the first two screens of Figure 2.

The system encompasses two primary components. The translation component identifies the elements that define every claim, i.e., candidates for datasets, attributes, rows, and comparison operations. The question planning component interacts with human domain experts to verify such elements and the checking results, optimizing verification tasks for maximal benefit.

Given claims C in a text document T, claims are verified in batches by human fact checkers. In each step, the algorithm selects an optimal batch N of claims for verification. Claim batches are selected based on multiple criteria, including expected verification overheads as well as their estimated utility for improving accuracy of the classifiers. For each selected claim in the current batch, we determine an optimal sequence of questions for the human checkers, minimizing expected verification time. Claims are validated or

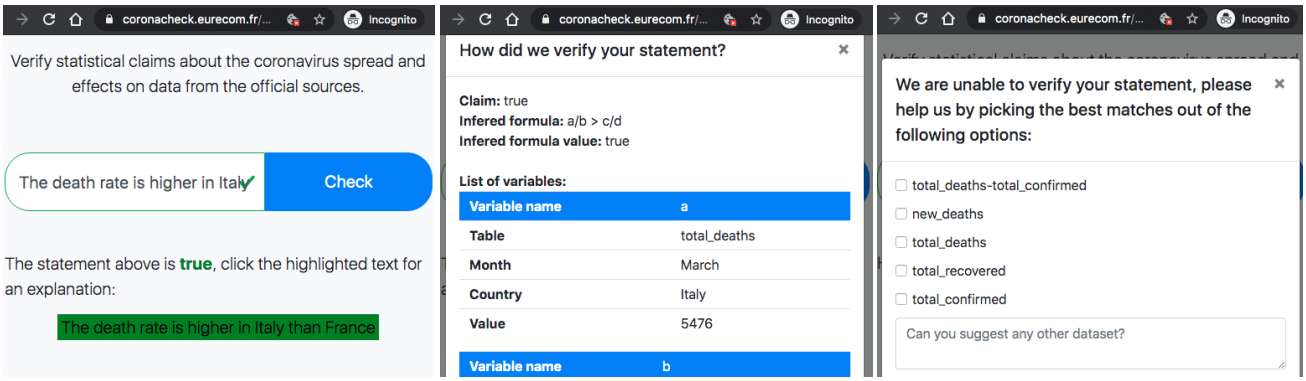


Figure 2: Screenshots of the demo for the check of a single claim (left) and its explanation (center). On the right, example of the feedback questions the system asks when model predictions have low confidence.

marked as erroneous, based on query evaluations. We remove the claims for which a verification result (i.e., either a verifying query or a decision that the claim is erroneous) can be calculated with sufficiently high confidence. Finally, the classifiers are retrained, based on the newly obtained classification results. We detail the two main components in the following.

2.1 Text to Query Translation

For a given claim, the system starts by executing classifiers over it to identify four elements. The first three are basic elements of every query: relevant relations, primary keys values (rows), and attributes names. The fourth classifier identifies a generic formula with variables in the place of keys and attribute values. This formula gets instantiated on the dataset at hand and becomes the combination of functions in the SELECT clause.

Example 2.1. Consider again the (false) claim "U.S. death rate is **1.3%** in March 18". The first classifier identifies that *Deaths* and *ReportedCases* relations can be used to verify it; the second classifier recognizes that rows reporting values for *U.S.A.* should be used; the third classifier returns *March 18* as the attribute of interest, and, finally, the fourth classifier returns the formula $\frac{a}{b}$. The output of the query is then compared to value 0.013 (1.3%) to assess the claim.

2.2 Question Planning

Our system relies on human fact checkers to verify generated translations of claims to queries. As soliciting feedback from workers is expensive, the question planning component uses cost-based optimization to determine effective question sequences. Question planning consists of two sub-tasks. First, for a fixed claim with low confidence in the models' predictions, we choose a sequence of questions to verify that claim with minimal expected overhead. Each question either solicits users to verify generated query elements, or to propose suitable elements themselves. Second, we decide the order in which claims are verified. When selecting claims to verify next, we take into account expected verification overheads as well as their value as training samples for our classifiers.

Example 2.2. Figure 2 (right) shows an example of a screen generated for a claim that could not be verified automatically with high confidence. Here, human workers are asked to select one out of multiple possible query aggregates. Depending on the amount of prior training data available, claims are typically verified by a sequence of such screens, focusing on different topics.

3 DEMONSTRATION

We consider two scenarios for our demonstration. First, we consider the "2018 World Energy Outlook Report" by the IEAwith associated data. Second, we demonstrate verification of claims related to the Coronavirus, using data sets published by organizations such as CDC (Center of Disease Control) and WHO (World Health Organization) as data sources.

We will prepare different demonstrations, targeted at visitors with different time budgets. First, visitors can get a quick impression of our system, without spending too much time, by applying it to single claims. Here, we will use the online version of our system for verifying claims about the coronavirus disease (COVID-19) outbreak. This version has already been trained for this domain, leveraging input by a large number of users as training data. Visitors can enter claims concerning the spread of the virus (e.g., "The death rate in Italy is much higher than in France." or "The total number of confirmed cases in USA remained constant from February to March"), and obtain a verification result as answer. Beyond the standard verification interface, we will show to interested visitors the queries into which claims are translated and will explain the translation process. Second, for visitors with more time, we will prepare a demonstration putting them into the roles of professional fact checkers. We will prepare extracts (one to two paragraphs) from the World Energy Outlook report and we will give visitors the opportunity to verify the extracts using our system. Here, visitors benefit from suggestions for data sets and query properties that are in most cases correct.

REFERENCES

[1] T. D. Cao, I. Manolescu, and X. Tannier. Searching for truth in a database of statistics. In *WebDB*, pages 4:1–4:6, 2018.
[2] S. Jo, I. Trummer, W. Yu, X. Wang, C. Yu, D. Liu, and N. Mehta. Verifying text summaries of relational data sets. In *SIGMOD*, pages 299–316, 2019.
[3] G. Karagiannis, M. Saeed, P. Papotti, and I. Trummer. Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification. *Proc. VLDB Endow.*, 13(11):2508–2521, 2020.
[4] G. Karagiannis, M. Saeed, P. Papotti, and I. Trummer. Scrutinizer: Fact checking statistical claims. *Proc. VLDB Endow.*, 13(12):2965–2968, 2020.

Human-in-the-Loop Schema Inference for Massive JSON Datasets (short version)

Mohamed-Amine Baazizi
Sorbonne Université, LIP6 UMR 7606
baazizi@ia.lip6.fr

Clément Berti
Sorbonne Université
clement.berti.upmc@gmail.com

Dario Colazzo
Université Paris-Dauphine, PSL Research University
dario.colazzo@dauphine.fr

Giorgio Ghelli
Dipartimento di Informatica, Università di Pisa
ghelli@di.unipi.it

Carlo Sartiani
DIMIE, Università della Basilicata
carlo.sartiani@unibas.it

ABSTRACT

JSON established itself as a popular data format for representing data whose structure is irregular or unknown a priori. JSON collections are usually massive and schema-less. Inferring a schema describing the structure of these collections is crucial for formulating meaningful queries and for adopting schema-based optimizations. In a recent work, we proposed a map-reduce schema inference approach that either infers a compact representation of the input collection or a precise description of every possible shape in the data. Since no level of precision is ideal, it is more appealing to give the analyst the freedom of choosing between different levels of precisions in an interactive fashion. In this paper we describe a schema inference system offering this important functionality.

1 INTRODUCTION

Borrowing flexibility from semistructured data models and simplicity from nested relational ones, JSON affirmed as a convenient and widely adopted data format for exchanging data between applications as well as for exporting data through Web API and/or public repositories. JSON datasets are usually retrieved from remote, uncontrolled sources, with partial, incomplete, or no schema information about the data. In these contexts, however, having a precise description of the structure of the data is of paramount importance, in order to design effective and efficient data processing pipelines. Schema inference, therefore, becomes a crucial operation enabling the formulation of meaningful queries and the adoption of well-known schema-based optimization techniques.

Several approaches and tools exist for inferring structural information from JSON data collections [11–13]. The common aspect of all these approaches is the extraction of some structural description with a precision that is fixed a priori, by the approach itself. While this methodology has the advantage of simplicity, it is in practice not satisfactory, since a JSON dataset can be rather (oftentimes highly) irregular in structure, and for this reason it can be typically described at different precision levels by a schema, while there exists no "best" precision level that can be fixed a priori. In general, one is interested in a description that is compact, easy to read even if it hides lots of details, typically in the first exploration steps, while in subsequent steps he/she is likely to be interested in a more precise, and therefore less succinct, schema description, where more details about the alternative shapes that can be found in the data are provided.

We believe that leaving the user the ability of tuning the level of precision of the inferred schema, by trying different possibilities and changing the level of details at different times, is an important feature, that existing techniques do not provide. With such a motivation in mind, in two recent works [8, 10], we devised, respectively, i) a MapReduce-based schema inference technique for massive JSON data that enables the user to chose, a priori, the level of precision of the inferred schema, and ii) a formal system which provides the user with mechanisms to interactively refine/expand the inferred schema, even locally, without the need of re-processing the data multiples times. The goal of this demonstration is to showcase results and mechanisms provided by these two works, by means of an implementation of the parametric schema inference system [8] which is based on Spark and which interacts with a Web interface that the user can exploit to choose or submit a dataset of interest, and to play with the interactive schema inference process [10].

The user interacts with the system by submitting a new dataset, or by choosing an existing, already analyzed, dataset. The system initially returns to the user a succinct schema with low precision level, and the user then can explore this schema in order to decide where to get more precision, at several nesting levels, by having the possibility of choosing a detailed schema description at a given level, while leaving the inner/nested levels described in a succinct fashion, hence at a lower degree of precision.

2 DEMONSTRATION OVERVIEW

The objective of this demonstration is to help the attendees understand the features and the goals of our schema inference system. To this aim, attendees will be able to:

(i) infer schemas for real-life JSON datasets according to K and L;

(ii) explore the inferred schemas and interactively fine-tune their precision;

(iii) get a concrete representation of the inferred schemas in JSON Schema.

2.1 System Architecture and Setup Details

Our system is based on a web application implemented following the client-server architecture depicted in Figure 1. The web client is used for loading the JSON collection and for performing the interactive schema inference while the web server is dedicated to storing the collection and to inferring the initial schema. The storage is supported by HDFS while the computation is ensured by Spark, as presented in [8]. The web client and the remote inference engine communicate through a REST API implemented in Python 3 using the Flask [2] library. The API requests from the client are processed by an orchestrator that coordinates between the storage and the inference modules in the server side. This coordination is ensured by API calls using two open source libraries: webHDFS [7] for communicating with the HDFS storage system, and livy [4] for submitting jobs to the Spark engine.

Upon receiving the input collection in JSONLines format [3], through the client, the server will store the collection on the HDFS then infers the L schema, using the Spark engine. The L schema is then sent to the client and used for inferring the K schema.The *schema visualizer* displays the K schema and translates the user actions into corresponding schema operations that are processed by the *schema manager*. The two modules coordinate during all the interaction session to fulfill the user requests.

The web client is implemented in Typescript [6] using the Angular 6 platform [1] which offers many advantages like modularity and the clean separation between the content of a web page and the program modifying its content; the schema inference of the server is implemented in Scala and is fully described in [8].

A lightweight system showcasing the core features of the full-system is available online at [5]. Differently from the full-system that will be demonstrated at the conference, the lightweight system performs schema inference on the client side and hence, it is limited to processing small size documents.

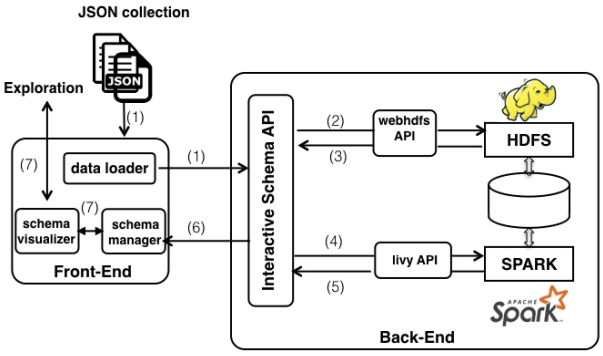


Figure 1: System architecture.

2.2 Demonstration Scenario

Our demonstration enables attendees to infer schemas for pre-loaded JSON datasets, to provide their own datasets, to explore the extracted schemas and to fine-tune their precision, as well as to convert them in more popular schema languages like JSON Schema. The datasets we plan to use in our demo are described below.

The GitHub dataset corresponds to metadata generated upon pull requests issued by users willing to commit a new version of code. It takes 14GB of storage and contains 1 million JSON objects sharing the same top-level schema and only varying in their lower-level schema. All objects of this dataset consist exclusively of records nested up to four levels of nesting. Arrays are not used at all.

The Twitter dataset corresponds to metadata that are attached to the tweets shared by Twitter users. It takes 23 GB of storage and contains nearly 10 million records corresponding, in most cases, to tweet entities. A tiny fraction of these records corresponds to a specific API call meant to delete tweets using their ids.

Finally, the NYTimes dataset contains approximately 1.2 million records and reaches the size of 22GB. Its records feature both nested records and arrays and are nested up to 7 levels. Most of the fields in records are associated to text data which explains the large size of this dataset compared to the previous ones.

3 RELATED WORK

The problem of inferring structural information from JSON received some attention as reviewed in our recent paper [9], outlining improvements of our approach w.r.t. state-of-the-art approaches for JSON schema inference, while the topic of interactive JSON schema inference was only recently addressed [10]. In the context of XML, the only work about interactive inference we aware of relies on user intervention for recognizing regular expressions that are similar enough to be merged and for deriving sophisticated XML schemas expressing complex constructs like inheritance and derivation [14].

REFERENCES

[1] Angular. available at `https://angular.io`.
[2] Flask. https://www.flaskapi.org.
[3] Jsonlines. http://jsonlines.org.
[4] livy rest api. available at `https://livy.incubator.apache.org`.
[5] Online demo. http://132.227.204.195:4200/host.
[6] Typescript. available at `https://www.typescriptlang.org`.
[7] webhdfs rest api. available at `https://hadoop.apache.org/`.
[8] M. A. Baazizi, D. Colazzo, G. Ghelli, and C. Sartiani. Parametric schema inference for massive JSON datasets. *VLDB J.*, 28(4):497–521, 2019.
[9] M.-A. Baazizi, D. Colazzo, G. Ghelli, and C. Sartiani. Parametric schema inference for massive json datasets. *The VLDB Journal*, pages 1–25, 2019.
[10] M. A. Baazizi, D. Colazzo, G. Ghelli, and C. Sartiani. A type system for interactive JSON schema inference (extended abstract). In C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 101:1–101:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.
[11] S. Scherzinger, E. C. de Almeida, T. Cerqueus, L. B. de Almeida, and P. Holanda. Finding and fixing type mismatches in the evolution of object-nosql mappings. In *Proceedings of the Workshops of the EDBT/ICDT 2016*, 2016.
[12] P. Schmidt. mongodb-schema, 2017. Available at `https://github.com/mongodb-js/mongodb-schema`.
[13] scrapinghub. Skinfer, 2015. Available at `https://github.com/scrapinghub/skinfer`.
[14] J. Vyhnanovska and I. Mlynkova. Interactive inference of XML schemas. In *Proceedings of the Fourth IEEE International Conference on Research Challenges in Information Science, RCIS 2010, Nice, France, May 19-21, 2010*, pages 191–202, 2010.

Clustering Massivement Distribué via Mélange de Processus de Dirichlet

Khadidja Meguelati
LIRMM, Inria, Univ Montpellier, CNRS, Montpellier, France
Montpellier, France
khadidja.meguelati@inria.fr

Bénédicte Fontez
MISTEA, Univ Montpellier, Institut Agro
Montpellier, France
bénedicte.fontez@supagro.fr

Nadine Hilgert
MISTEA, Univ Montpellier, INRAE
Montpellier, France
nadine.hilgert@inrae.fr

Florent Masseglia
LIRMM, Inria, Univ Montpellier, CNRS, Montpellier, France
Montpellier, France
florent.masseglia@inria.fr

Isabelle Sanchez
MISTEA, Univ Montpellier, INRAE
Montpellier, France
Isabelle.Sanchez@inrae.fr

RÉSUMÉ

La classification non supervisée (ou clustering) a pour objectif d'identifier des classes pertinentes dans les données. Elle est largement utilisée dans de nombreuses applications telles que le marketing, la reconnaissance de patterns, l'analyse de données et le traitement d'images. Déterminer le nombre optimal de clusters dans un ensemble de données est un défi fondamental qui a ouvert de nombreuses directions de recherche et a vu de multiples méthodes proposées pour y répondre.

Le Mélange de Processus de Dirichlet (DPM) est utilisé pour le clustering car il permet de définir automatiquement le nombre de classes, mais les temps de calculs qu'il implique sont généralement trop importants, nuisant à son adoption et rendant inefficaces ses versions centralisées.

Nous abordons la parallélisation du mélange de processus de Dirichlet pour améliorer ses performances en exploitant des environnements massivement distribués. En effet, d'après la littérature, l'algorithme de DPM distribué fait appel à de nombreux problèmes tels que : l'équilibre de charge entre les nœuds de calcul, les coûts de communication, et le plein bénéfice de propriétés du DPM.

Cette démonstration concerne deux nouvelles approches pour le clustering parallèle via DPM : *i)* DC-DPM (Clustering Distribué via mélange de processus de Dirichlet) [1], une version parallélisée, qui permet le clustering de millions de points de données, ce qui représente un vrai défi. *ii)* HD4C (Clustering de Dirichlet Distribué pour des Données de Haute Dimension) [2], une solution de clustering parallèle qui s'adresse au problème de la dimensionnalité qui devient un défi important avec les obstacles numériques et théoriques dans ce cas. La première s'adapte à des données massives en exploitant les architectures distribuées. La deuxième ajoute le clustering de données de haute dimension telles que les séries temporelles (en fonction du temps), les données hyperspectrales (en fonction de la longueur d'onde), etc.

DC-DPM et HD4C sont implémentés avec Spark et le code est mis à disposition ici : https://github.com/khadidjaM.

L'interface graphique de la démonstration est disponible ici : http://147.100.179.112:3838/team/kmeguelati/dpmclustering/. La vidéo de la démonstration est disponible à : https://drive.google.com/file/d/1GHLF5csHk8Oa7PZK4dwA3RTK35KNIbne/view?usp=sharing

RÉFÉRENCES

[1] Khadidja Meguelati, Bénédicte Fontez, Nadine Hilgert, and Florent Masseglia. 2019. Dirichlet Process Mixture Models made Scalable and Effective by means of Massive Distribution. In *SAC : Symposium on Applied Computing*. Limassol, Cyprus. https://doi.org/10.1145/3297280.3297327

[2] Khadidja Meguelati, Bénédicte Fontez, Nadine Hilgert, and Florent Masseglia. 2019. High Dimensional Data Clustering by means of Distributed Dirichlet Process Mixture Models. In *IEEE International Conference on Big Data (IEEE BigData)*. Los-Angeles, United States. https://hal-lirmm.ccsd.cnrs.fr/lirmm-02364411

EPIQUE: Extracting Meaningful Science Evolution Patterns from Large Document Archives

Ke Li
LIP6, CNRS, Sorbonne Université
Paris, France
ke.li@lip6.fr

Hubert Naacke
LIP6, CNRS, Sorbonne Université
Paris, France
hubert.naacke@lip6.fr

Bernd Amann
LIP6, CNRS, Sorbonne Université
Paris, France
bernd.amann@lip6.fr

Science evolution can broadly be studied by adopting a cognitive view or a social view on the evolution dynamics. The cognitive view emphasizes the shared knowledge and the change of ideas (Kuhn's approach [4]), whereas the social view takes account of authorship and social interaction (e.g., citation graphs) [2, 6]. Bibliographic archives often include both kinds of information and there also exist methods which combine both views to study science evolution [3]. In the interdisciplinary EPIQUE project[1] we adopt the cognitive view for modeling science evolution and assume that the evolution only depends on the document contents. Whereas this choice clearly reduces the expressivity of our evolution model it also decreases the "social" bias and detects more easily possible interactions between scientific ideas and contributions independently of any particular scientific community. The goal of our work described in [5] is to develop a general framework for exploring the evolution of science in document archives. The main contributions of this work are the following [5]:

- We propose a generic topic evolution model enabling the specification and extraction of meaningful topic evolution patterns called *topic pivot graphs*.
- We define high-level measures for estimating the quality of the topic extraction process and for characterizing the structural and quantitative evolution of topics during a time period. This enables the experts to tune the topic extraction process and explore large topic evolution graphs by defining complex topic evolution patterns.
- We implemented a scalable prototype on top of Apache Spark for processing large scientific corpora containing millions of documents and finding meaningful topic evolution graphs for both stable topics and highly evolving ones.

The goal of this demonstration is topresent an implementation of this framework on top of Apache Spark.

Architecture overview. Figure 1 illustrates the overall workflow which takes as input a corpus of documents split into several, possibly overlapping time periods (the same document might appear in two periods). All documents within a period are processed by LDA [1] to generate a set of topics which are aligned to produce a global topic evolution graph G_{β_0}. This global evolution graph is then decomposed into n families of topic evolution patterns, also called *pivot evolution graphs*, defined by a set of topic alignment

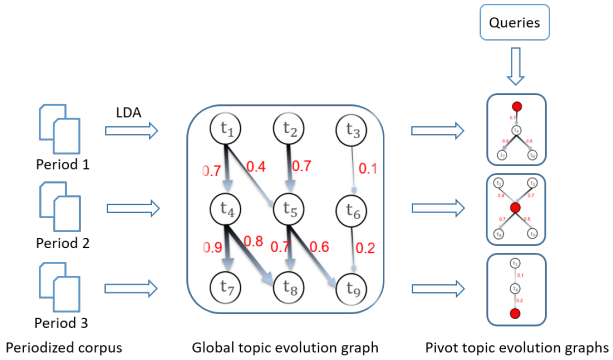


Figure 1: Topic evolution model of EPIQUE

thresholds $\beta_i > \beta_0$, $1 \leq i \leq n$. These graphs can then be queried using the filters defined in [5].

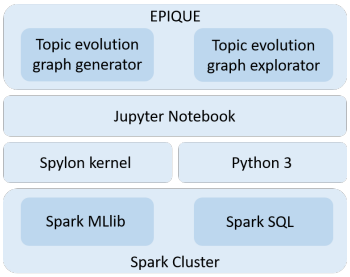


Figure 2: EPIQUE web application architecture

Figure 2 gives an overview of the architecture of our web application implemented on top of Apache Spark and Jupyter Notebook. The entire process to study science evolution over a corpus is split into two steps for building the pivot evolution graphs and for interactively exploring these graphs. Each step corresponds to a separate user interface. The evolution graph generation is implemented in Scala and exectued through the Spylon[2] kernel. Evolution graph exploration uses a standard Python kernel to take advantage of advanced Python 3 graphical user interface libraries for facilitating user interaction.

Demonstration Scenarios. Our EPIQUE prototype allows the audience to easily and intuitively generate high-quality evolution graphs and explore them. Among the corpora we have prepared

[1]This work was funded by French ANR-16-CE38-0002-01 project EPIQUE.

[2]https://github.com/Valassis-Digital-Media/spylon-kernel

in several domains, our demonstration focuses on the evolution of computer science based on the arXiv corpus (1156114 documents, 10 3-year time periods (1 year overlap) covering the publications from 1998 to 2017 and 100 topics / period). We propose two interactive demonstration scenarios[3].

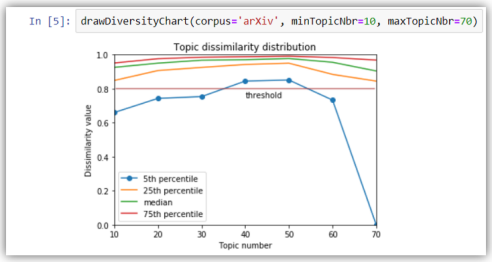


Figure 3: Screenshot: topic diversity evaluation

Scenario 1 The audience selects or uploads a corpus of documents with a vocabulary of terms pre-processed by an on-line text-mining tool Gargantext[4] and specifies the time periods through sliding window over a global time period. Then, the LDA topic model is generated for each period. LDA requires a vocabulary and a number of topics to be generated. This number obviously influences the diversity of the resulting topics. Therefore, the application first generates a set of topic models for different topic numbers per period. The user can then visualize the diversity of the extracted topic models (topic dissimilairty distribution) and choose the model with the highest diversity for each period. A topic diversity distribution for different topic numbers is reported as shown in Figure 3 and, for example, by observing the 5th percentile values (blue line), the user can retain one of the two models (40 or 50 topics per period) that achieves 95% of pairwise dissimilarities above 0.8.

Then, the topics of consecutive periods are aligned and all pivot topic evolution graphs are generated along with their main temporal, structural and evolution indicators: *liveliness*, *split degree*, etc. All topic labels are also generated automatically in this step.

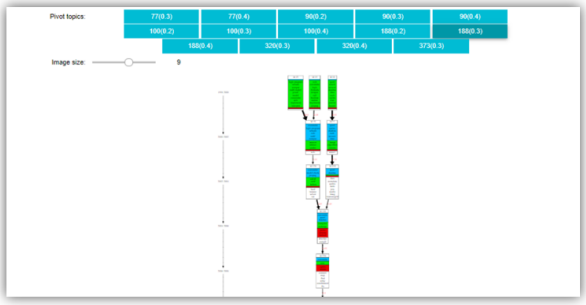


Figure 4: Screenshot: pivot topic evolution graph visualization

In the next step, the user specifies its exploration goal through an intuitive declarative query-by-example interface (as shown in the demonstration video[3]) and visualizes pivot topic evolution graphs as shown in Figure 4.

We showcase a search for topic graphs containing a given term (*e.g.*, database) or set of terms suggested by the audience. Besides the topic content, the audience can search for topic graphs on their shape as well. We also prepared 10 predefined query templates for a typical shapes of high interest such as (i) topics that split in distinct 5+ years long branches, (ii) topic graphs with low *relative evolution degree* and high end-to-end *pivot evolution degree*. We also demonstrate more complex queries combining several query templates to build, for example, *concept drift* queries looking for pivot topics that contain emerging terms originating from other, "older" topics which are not part of their past pivot subgraph.

Scenario 2 In the second scenario, we will provide the audience with the possibility to prepare their own corpus using the Gargantext service which is also part of the EPIQUE project. Gargantext includes a number of bibliographic archives like Pubmed, Web of Science, etc. and allows to create domain specific document collections and vocabularies which are then processed by the same workflow as in Scenario 1.

Future Work. In the next step, we intend to optimize the computation of pivot topic evolution graphs and exploit the LDA document-topic matrix for enriching the analysis. Additionally, we plan to integrate other topic extraction methods than LDA. This prototype will also be used to validate our evolution model with philosophers of science to define and extract complex evolution patterns from different scientific domains.

[3] see http://www-bd.lip6.fr/wiki/site/recherche/projets/epique/demo/start for a video demonstration

[4] https://gargantext.org/

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[2] Eugene Garfield. 1955. Citation Indexes for Science: A New Dimension in Documentation through Association of Ideas. *Science* 122, 3159 (1955), 108–111.

[3] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and Lee Giles. 2009. Detecting Topic Evolution in Scientific Literature: How Can Citations Help?. In *ACM Conference on Information and Knowledge Management*. ACM, 957–966.

[4] Thomas S. Kuhn, Otto Neurath, and Thomas Samuel Kuhn. 1994. *The Structure of scientific revolutions* (2nd ed., enlarged ed.). Number ed.-in-chief: Otto Neurath ; Vol. 2 No. 2 in International encyclopedia of unified science Foundations of the unity of science. Chicago Univ. Press, Chicago, Ill.

[5] Ke Li, Hubert Naacke, and Bernd Amann. 2020. EPIQUE : A Graph Data Model and Query Language for Exploring the Evolution of Science. In *36ème Conférence sur la Gestion de Données – Principes, Technologies et Applications (BDA 2020)*.

[6] Xiaoling Sun, Jasleen Kaur, Staša Milojević, Alessandro Flammini, and Filippo Menczer. 2013. Social Dynamics of Science. *Scientific Reports* 3 (2013), 1069.

7 Résumés des articles de doctorant

SLA Definition for Multi-Cloud Queries

Damien T. Wojtowicz
damien.wojtowicz@irit.fr
IRIT – Université Toulouse III

Shaoyi Yin
shaoyi.yin@irit.fr
IRIT – Université Toulouse III

Franck Morvan
morvan@irit.fr
IRIT – Université Toulouse III

ABSTRACT

Public data availability in cloud-hosted databases raises interests in systems providing multi-cloud querying capabilities. Since data access in this context induces monetary costs, we suggest a method to compute SLAs for multi-cloud queries. It consists in decomposing queries into a directed graph of maximal sub-queries per provider, and finding a financially cheapest execution plan. This method yields SLAs with monetary costs lower than a download-all approach, granted that inter-provider data transfers are minimised.

KEYWORDS

Database-as-a-Service, Multi-Cloud, Service-Level Agreement.

1 INTRODUCTION

Cross-analysis of datasets is of key importance to researchers, as the accumulation of data in fields such as astronomy and biology dramatically increases the opportunities for new discoveries. Various data sharing methods are available. File provision, either through a web interface or in the cloud[1], is popular and historically ingrained. Sharing can also be achieved by offering public read access to a database, which may be hosted in the cloud following the Database-as-a-Service (DBaaS) model[2]. In this case, users can leverage the services of public cloud providers in order to execute their queries. Alternatively, users can download datasets and run queries on self-administrated databases, implying sufficient skills and resources.

Unfortunately, providers do not offer out-of-the-box multi-cloud querying capabilities, leaving space for systems orchestrating query execution across multiple public databases in a DBaaS fashion. Since access to the providers' datasets is billed, such a service shall let users control their expenditures by means of Service-Level Agreement (SLA). Those contracts are known to play a key role in query optimisation [8], hence the importance of well-defined SLAs. In the context of DBaaS, a single performance objective cannot be retained for all queries due to different complexities. In addition, even for the same query, different tenants may have different performance expectations which are partially influenced by their budget (i.e. minimising response time may not be the goal). These challenges should be taken into account in the SLA definition process.

In Section 2, we briefly review literature about multi-cloud database systems. In Section 3, we propose a graph-based SLA definition method for multi-cloud relational queries. Its results are analysed in Section 4. We conclude in Section 5 by mentioning perspectives.

[1]See examples of public datasets at Amazon (https://registry.opendata.aws/) or Google (https://cloud.google.com/public-datasets)
[2]See examples at Google (https://cloud.google.com/bigquery/public-data).

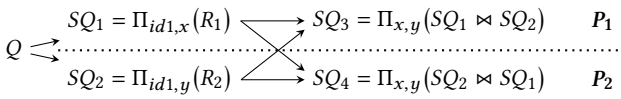


Figure 1: DAG and minimal sub-queries $SQ_\star$ for query Q. Two execution plans, both involving data transfer between providers P_1 and P_2, are possible with SQ_3 and SQ_4.

2 RELATED WORK

Work on multi-cloud database services has mostly focused on federated databases. Indeed, several query processing systems involving different cloud providers exist. To name a few, SCOPE [5] is the base of a collaborative document editing tool, MetaStorage [2] operates as a key-value storage system and SHAMC [6] acts as a relational DBMS. Despite their seemingly differences, in terms of context, objectives and data models, they share the same approach.

Indeed, they all own their data, seeking an overall system optimisation, using for example data placement or replication strategies, with respect to an objective (e.g. response time, availability, financial cost) rather than solely focusing on optimising data access. These systems are based on the Infrastructure-as-a-Service (IaaS) model, and their SLAs encompass broader services than querying.

Research has also been carried on multi-objective cost models for multi-cloud queries [3], aiming at adding a monetary aspect to usual cost models. In this case, query execution is performed on a set of cloud-hosted virtual machines leveraging IaaS capabilities. Those cost models are best suited to scale up or down resources, and are therefore not suitable to help defining SLAs for DBaaS-based multi-cloud systems.

3 SLA DEFINITION METHOD

In this paper, we focus on the performance objective for a query and its relation to the monetary costs. Therefore, we model the SLA for a query as a couple $SLA =< C, RT >$, with C the monetary cost of a query and RT its response time. This paper is exemplified by the query $Q = \Pi_{x,y}(R_1 \bowtie R_2)$, assuming relations $R_1(\underline{id1}, x)$ and $R_2(\underline{id2}, \#id1, y)$ respectively hosted on providers P_1 and P_2 (see fares[3] in Table 1a). For the sake of simplicity, network bandwidth in our example is supposed to be constant at 1.0 GB/s.

Our method is a three-step procedure, starting by decomposing the input query, then generating SLAs for sub-queries and ending by aggregating the latter.

[3]Inspired from BigQuery's pricing policy (https://cloud.google.com/bigquery/pricing)

3.1 Query decomposition

Inspired by query decomposition techniques [3, 7], we suggest that multi-cloud queries can be modelled as a directed acyclic graph (DAG), where vertices are maximal sub-queries involving a minimal set of providers. Query Q can be decomposed as depicted in Figure 1. SQ_1 and SQ_2 are maximal sub-queries involving a single provider, respectively P_1 and $P2$. SQ_3 and SQ_4 involve data from P_1 and P_2 respectively executed on P_1 and P_2.

This decomposition creates two plans, one ending with SQ_3 and the other with SQ_4. It is worth noticing that both are similar to execution plans in distributed databases. In this context, the best plans usually minimise network transfers [9].

3.2 Mitigation of provider miscalculations

For each sub-query $SQ_\star$, a SLA is generated as well as an estimation of the output relation's size S (see Table 1d). Those might not reflect the actual sub-queries' output relation size due to internal cardinality estimation errors [4]. We suggest to keep track of those estimation errors by computing a supposed error ratio for each query submitted to the provider as $r = \frac{S^{(Real)}}{S^{(SLA)}}$.

Those ratios are then used to compute $\overline{r}$, a sliding average of r on the last n queries exemplified in Table 1f. While being sensitive to the complexity of the last queries, using a sliding average let our system take into account changes of the provider's estimator.

3.3 SLA components aggregation

We use separate methods for each component of the SLA, using corrected values (see Table 1e). Monetary cost C is the sum of all sub-queries costs and all transfers costs (see the latter in Table 1b). Response time RT is the sum of the RT-maximal path in the DAG, taking into account cross-providers transfer time.

Due to space limitation, we cannot put the cost formulas in the paper. We only show the results for illustration purposes. Table 1c shows the costing for each plan. At the end of the procedure, the SLA that will be presented to the user, in the context of our example query Q, is $< 0.184\,\$, 7.287\,s >$ stemming from SQ_4.

4 RESULT ANALYSIS

Figure 2 depicts the breakdown of each scenario costs. The less expansive one minimises inter-provider transfers, which is unsurprising given the similarity of our setting with distributed systems. Moreover, there is no significant differences between SQ_3 and SQ_4 in storage and processing costs.

SQ_4 is 1.8 times cheaper with an acceptable performance degradation (43%) compared to the download-and-process scenario, thus a multi-cloud query appears to be more competitive financially-wise than the download-all approach.

5 CONCLUSION

Our SLA computation method is a first step towards a middleware enabling multi-cloud querying in a DBaaS fashion. We showed that optimal multi-cloud query execution plans should minimise inter-providers transfers in order to limit costs.

Next steps will involve setting up a SLA-constrained execution engine. Given that sub-queries' output relation size is an estimation, the SLA-based optimal plan might not actually be the best. Hence, methods mitigating estimation errors, such as mobile-agent-based models [1] implementing reinforcement learning, appear to be relevant for the execution of multi-cloud queries.

Table 1: Values used for SLA computations

	P_1	P_2
Export	0.060	0.075
Querying	0.006	0.008
Storage	0.003	0.002

(a) Providers' fares ($/GB)

Transfer	Time (s)	C ($)
$SQ_1 \rightarrow SQ_4$	1.102	0.066
$SQ_2 \rightarrow SQ_3$	3.050	0.229

(b) Transfer times and fares

Scenario	C ($)	RT (s)
SQ_3	0.353	10.762
SQ_4	0.184	7.287
Download-all	0.335	5.083

(c) Costs of all scenarios

$SQ_\_$	C ($)	RT (s)	S (GB)
SQ_1	0.009	1.000	1.000
SQ_2	0.030	2.000	3.000
SQ_3	0.055	6.000	6.100
SQ_4	0.061	5.000	6.120

(d) Provider-generated SLA of each sub-query

$SQ_\_$	C ($)	RT (s)	S (GB)
SQ_1	0.010	1.102	1.102
SQ_2	0.031	2.033	3.050
SQ_3	0.060	6.610	6.720
SQ_4	0.062	5.083	6.222

(e) Corrected SLAs

Query	1	2	3	4	5	6	$\overline{r}$
r for P_1	1.50	0.90	1.20	1.05	1.01	0.95	**1.102**
r for P_2	1.01	1.02	0.95	1.03	1.10	0.99	**1.017**

(f) Last SLA estimation error ratio for providers and $\overline{r}$ computed using the last $n = 6$ queries for each provider

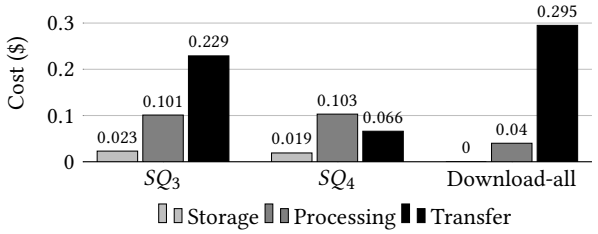


Figure 2: Breakdown of each scenario cost. Inter-providers transfers can tremendously increase the price of a scenario.

REFERENCES

[1] J.-P. Arcangeli, F. Morvan, A. Hameurlain, and F. Migeon. 2004. Mobile Agents Based Self-Adaptive Join for Wide-Area Distributed Query Processing. *Journal of Database Management* 15, 4 (2004), 25–44.

[2] D. Bermbach, M. Klems, S. Tai, and M. Menzel. 2011. MetaStorage: A Federated Cloud Storage System to Manage Consistency-Latency Tradeoffs. *IEEE CLOUD*, 452–459. Washington, USA.

[3] A. Gounaris, Z. Karampaglis, A. Naskos, and Y. Manolopoulos. 2014. A Bi-Objective Cost Model for Optimizing Database Queries in a Multi-Cloud Environment. *Journal of Innovation in Digital Ecosystems* 1, 1 (2014), 12–25.

[4] V. Leis, A. Gubichev, A. Mirchev, P. Boncz, A. Kemper, and T. Neumann. 2015. How Good Are Query Optimizers, Really? *Proc. VLDB Endow.* 3, 9 (2015), 204–215.

[5] A. Rafique, D. van Landuyt, E. Truyen, V. Reniers, and W. Joosen. 2019. SCOPE: Self-Adaptive and Policy-Based Data Management Middleware for Federated Clouds. *Journal of Internet Services and Applications* 10, 1 (2019), 2.

[6] L Wang, Z Yang, and X Song. 2020. SHAMC: A Secure and Highly Available Database System in Multi-Cloud Environment. *FGCS* 105 (2020), 873–883.

[7] E. Wong and K. Youssefi. 1976. Decomposition — a Strategy for Query Processing. *ACM TODS* 1, 3 (1976), 223–241.

[8] S. Yin, A. Hameurlain, and F. Morvan. 2018. SLA Definition for Multi-Tenant DBMS and its Impact on Query Optimization. *IEEE TKDE* 30, 11 (2018), 2213–2226.

[9] M. T. Özsu and P. Valduriez. 2020. *Principles of Distributed Database Systems* (4 ed.). Springer International Publishing.

Adaptive Search Engine for Heterogeneous Documents

Oussama Ayoub
oayoub@sevillemore.com
SMH & Léonard de Vinci Pôle Universitaire, Research Center
Paris La Défense, France

Christophe Rodrigues
christophe.rodrigues@devinci.fr
Léonard de Vinci Pôle Universitaire, Research Center
Paris La Défense, France

Nicolas Travers
nicolas.travers@devinci.fr
Léonard de Vinci Pôle Universitaire, Research Center
Paris La Défense, France

ABSTRACT

Providing an efficient search engine for legal actors querying for textual documents is a challenging objective. Nowadays most engines target semantic analysis on top of text queries to enhance the relevance. But the legal context relies mainly on heterogeneous data in terms of both queries and documents length, structural complexity, and queries context. This combination makes standard solutions hardly scalable or adaptable. The proposed solution is an adaptive approach that aims to be applied to any textual database establishing a search engine. Its peculiarity is to normalize documents by producing fragments, enriching them with word embedding, here summarizing, and rebuilding documents through similarity aggregations on either enriched content, structure and context. By integrating our solution in Elasticsearch we ensure the flexibility and the fine-tuning of both words embedding and similarities.

KEYWORDS

Natural Language Processing, Information Retrieval, Search Engine

1 INTRODUCTION

The legal context relies on several types of documents which makes the comparison between them a real issue and consequently complexifies the establishment of a search engine. Moreover, several major constraints must be considered to deliver a relevant service that is fully integrated into the legal environment: professional secrecy, the user's legal context and the consideration of case law.

Providing dedicated solutions for each data type is counterproductive. Thus, the issue is to propose a unique solution which makes all data comparable regardless of its size (documents and queries), structure (contracts, profiles), richness (from simple to complex documents), or context of use (simple search to recommendations. On top of that, relevance must be tuned according to a given context, here the legal one. The main issue to tackle is to deal with really heterogeneous documents length from a single paragraph to hundreds-page documents with a complex structure and consequently its correlation with relevance and similarity measures. We will detail this problematic in the following.

Our approach combines two different domains, linked to Natural Language Processing (NLP) and a mix between Information Retrieval (RI) and Databases (DB), into an elaborate architecture that provides access to an enhanced search engine. It benefits from the NLP modeling in order to provide summaries on data according to the context (*i.e.,* local vs. global corpus content) and from the IR/DB capabilities to manipulate results to reconstitute documents and to make them comparable and scalable.

This paper aims at giving the subject's relation to the literature and an overview on the approach's architecture. Section 2 describes the main issues related to the subject and the related work on those topics. Section 3 presents the approach and the global architecture of the search engines designed to answer the preceding issues. Finally, we conclude in Section 4 with current paths of research.

2 PROBLEMATIC AND STATE-OF-THE-ART

The goal of this research topic is to create an adaptive search engine that can be applied on dedicated environments (*i.e.,* legal "clusters"), associated with external resources and knowledge. We are facing issues: (1) various documents length (not compatible with the state-of-the-art on preprocessed representations of documents), (2) queries length varies from keywords to complex documents (relevance is barely dealt with content inclusion), (3) target context-aware relevance (*i.e.*, legal data). Thus, the global approach mainly targets this goal by trying to make various documents and queries comparable. Consequently, our work meets two research domains and mix them in a relevant and efficient way: Natural Language Processing (NLP) and Information Retrieval (IR).

Language processing, based on learning models, aims to categorize legal issues. The association of different legal codes makes Topic Modeling and Abstract summary generation main objectives to address. In 2013, [3] proposed the creation of word representation in vector space known as `Word2Vec`. It changed the perspective in NLP by adding the context of words in their vectorized representation. A lot of work has been done since to create more significant representations by using attention models. Encoders, autoencoders and later transformers (combining few technologies) are many ways of creating a complex and efficient vector to represent and store the data in a small dimension. Some papers are using embeddings as representation of the data object to enable searching documents by making them comparable. Jan Rygl and al. [5] propose to transform obtained embedding with Latent Semantic Analysis into strings allowing the use of full-text search using TF-IDF similarity in ElasticSearch built with the default configuration. They refined the query result by calculating the cosine similarity between vectors on remaining documents. [4] used a neural autoencoder to create representation of documents to find similarities between different types of objects in their search engine. The combination of multiple techniques including autoencoders improves the quality of results. We differ from those approaches by introducing preprocessing steps to normalize heterogeneous documents analysis and aggregations in post-processing to rebuilt documents using similarities.

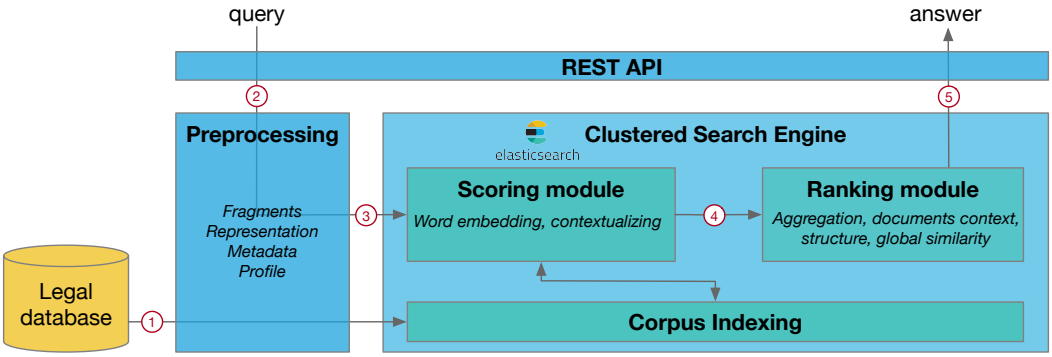


Figure 1: Architecture

Some studies combine IR with Machine Learning to facilitate the analysis of legal corpora while providing new functionalities to define a dedicated legal search engine. The issue on documents length and heterogeneity is ignored or avoided when training the main Machine Learning and Deep Learning solutions nowadays [1, 2] while documents in legal domain have a much higher length. This fact was recently raised [1, 6] where well-known machine learning methods do not perform as well as expected for long-length documents. [6] tries to solve this problem by separating the documents in chunks and merge their vectors to create the document's vector. Moreover, the conversion process is applied to queries as well as for documents without taking the length into account [4]. Those solutions inspired to build our approach by generalizing the concepts of embedding on various documents and queries adaptable on dedicated contexts.

3 APPROACH

We propose a full architecture that solves the issues mentioned. Figure 1 shows the mandatory modules for the operation of the search engine. The processing steps are detailed in the following:

`Step 1` - The contextual database is pre-processed by a refinement module in order to compute AI models and document transformations. This crucial step normalizes documents' size either by splitting in nominal sizes called "*fragments*" as well as enriching its content with "*word embedding*" according to the context of use. We intend to combine fragmentation and summarize processes to enhance the quality of relevance at various scales. Moreover, structural and context information of documents enrich the embedding, which constitutes another dimension of relevance in a hyper-connected environment where most of the documents are linked.

`Step 2` - Every query is pre-processed to be converted similarly as the refined stored documents. This request can be of various lengths, type or linked to a user's profile.

`Step 3` - Document fragments are projected in a vector space model relying on both stored data and the IA model in order to give a score for each fragment. The scoring is also based on plain text similarity to maximize the relevance of the results. By integrating the approach into `Elasticsearh`, it allows to make the computation scalable and tunable. Thus, the tuned scoring computation is distributed on each fragment by reducing the information of a document to one vector. It returns a detailed explanation of the result used to recompose documents.

`Step 4` - Relevant fragments are grouped together to give a global and contextual answer. This aggregation step is a major issue of the overall system since it allows targeting various types of use cases by mixing the combination of aspects in the vector space.

`Step 5` - The answer is a set of documents composed of linked fragments with enriched explanations of provided scores.

4 CONCLUSION

We introduced a new flexible architecture establishing a powerful search engine relying on semantic, structural and context analysis. Our work focuses on solving the heterogeneity of databases' documents, especially in terms of length, using the segmentation of the documents into fragments to enhance the relevance of results.

The design of this solution represents the initial step of our work. As future work, we will focus on confirming the key assumptions regarding the benefits of word embedding, segmentation and aggregation in search engines as well as combining multiple scoring criteria on different datasets (*e.g.*, legal, tourism, Kaggle).

ACKNOWLEDGMENT

This work has been funded by the research agreement "Smart Legal" in collaboration with *Seville More Helory*.

REFERENCES

[1] Robert Keeling, Rishi Chhatwal, Nathaniel Huber-Fliflet, Jianping Zhang, Fusheng Wei, Haozhen Zhao, Ye Shi, and Han Qin. 2019. Empirical Comparisons of CNN with Other Learning Algorithms for Text Classification in Legal Document Review. *2019 IEEE International Conference on Big Data (Big Data)* (2019).

[2] Aurelie Mascio, Zeljko Kraljevic, Daniel Bean, Richard Dobson, Robert Stewart, Rebecca Bendayan, and Angus Roberts. 2020. Comparative Analysis of Text Classification Approaches in Electronic Health Records. *CoRR* (2020).

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR'13 Workshop*.

[4] Jonas Pfeiffer, Samuel Broscheit, Rainer Gemulla, and Mathias Göschl. 2018. A Neural Autoencoder Approach for Document Ranking and Query Refinement in Pharmacogenomic Information Retrieval. In *BioNLP workshop of ACL*.

[5] Jan Rygl, Jan Pomikálek, Radim Rehurek, Michal Ruzicka, Vít Novotný, and Petr Sojka. 2017. Semantic Vector Encoding and Similarity Search Using Fulltext Search Engines. *CoRR* abs/1706.00957 (2017).

[6] Lulu Wan, George Papageorgiou, Michael Seddon, and Mirko Bernardoni. 2019. Long-length Legal Document Classification. arXiv:1912.06905 [cs.CL]

8 Prix BDA 2020

8.1 Prix des articles de recherche

BDA a la particularité de proposer deux catégories d'articles : les articles originaux non publiés et les articles publiés récemment dans une conférence internationale de renom. Cette dernière catégorie permet de diffuser largement les travaux faisant la renommée internationale de notre communauté nationale en gestion de données.

Lauréats du prix des articles de recherche

Article non publié : A Partitioning Approach for Skyline Queries in Presence of Partial and Dynamic Orders, *Karim Alami and Sofian Maabout*

Article non publié : Not Elimination and Witness Generation for JSON Schema, *Mohamed-Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani and Stefanie Scherzinger*

Article publié à ICDT'20 : Une dichotomie sur l'évaluation de requêtes closes sous homomorphismes sur les graphes probabilistes, *Antoine Amarilli et Ismail Ilkan Ceylan*

Article publié à PVLDB'20 : Guided Exploration of User Groups, *Mariia Seleznova, Behrooz Omidvar-Tehrani, Sihem Amer-Yahia et Eric Simon*

8.2 Prix des démonstrations

Lauréat du prix des démonstrations

Scrutinizer : A System for Checking Statistical Claims, *Georgios Karagiannis, Mohammed Saeed, Paolo Papotti and Immanuel Trummer*

8.3 Prix des thèses en gestion de données

Lauréats du prix des thèses

Prix de thèse : Maxime Buron pour sa thèse intitulée
« *Efficient Reasoning on Large and Heterogeneous Graphs* ».

Accessit au Prix de Thèse : Chao Zhang pour sa thèse intitulée
« *Optimization of User-Defined Aggregate Functions : Parallelization and Sharing* »